OmegaT 3.1 - 取扱説明書

Vito Smolej [FAMILY Given]

OmegaT 3.1 - 取扱説明書

Vito Smolej [FAMILY Given]

発行日

概要

この文書は　OmegaT（自由に使用できる、コンピューター翻訳支援ツール）の公式な取扱説明書です。インストール手順の説明も、含まれています。

目次

図の一覧

表の一覧

第1章 OmegaT について - はじめに

1. OmegaT の特徴

OmegaT は、自由に使用できる、マルチプラットフォームのコンピューター翻訳支援ツールです。主に以下のような特徴があります：

- 翻訳メモリ：OmegaT は、単語や分節をどのように翻訳したかを翻訳メモリに記憶していきます。同時に、それまでに記憶した翻訳内容をメモリから参照することができます。翻訳メモリは、同じ分節の繰り返しや、似通った表現が多く現れる翻訳に非常に便利です。OmegaT は翻訳メモリを使用して、以前行った翻訳内容を記憶し、翻訳しようとしている文章にもっとも適した翻訳の提案を行います。

  翻訳メモリは、以前翻訳した文書を更新するときに非常に便利です。変更されていない文章は、自動的に翻訳されます。更新された文章は未翻訳として扱われますが、翻訳済みの文章の中から翻訳候補が表示されるようになります。従って、元の文書の翻訳修正という点でも、とても簡単に扱えます。たとえば、翻訳会社や顧客から以前の翻訳メモリを受け取った場合、OmegaT はそれを参照用として使用できます。

  OmegaT が使用する翻訳メモリファイルは標準的な TMX フォーマットです。このファイル形式をサポートする他の翻訳ツールであれば、翻訳資産を相互使用できます。

- 用語管理：翻訳の一貫性を保つ上で、用語管理は重要です。OmegaT には用語集（glossary）の機能があり、（たとえば特定分野向けの）2 言語間の対訳集として、1 単語または短いフレーズの対訳を翻訳中に参照することができます。現在翻訳している文章中に、用語集に登録済みの単語があった場合は、参照用として用語集ウィンドウにその対訳が表示されます。

- 翻訳作業の流れ：翻訳対象は、ファイル 1 つのこともあれば、いろいろな形式のファイルを含んだたくさんのフォルダー、という場合もあるでしょう。OmegaT は、指定された翻訳対象ファイル群をファイルフィルター機能によって識別し、各ファイルのテキスト部分を抽出します。さらに、分節化機能によってそのテキスト部分を複数のまとまりに分割し、翻訳作業を行いやすいように順に表示します。翻訳中は、その翻訳内容を記憶すると共に、翻訳メモリからよく似た分節を探し出して提示します。翻訳し終えたら、訳文ファイルを生成して、適切なプログラムで開き、翻訳結果を完成形で見る…という具合です。

2. 各章の概要

この説明書は、チュートリアルとリファレンス両方の役割を兼ねることを意図しています。ここで簡単に、各章の概要をご紹介しましょう。

- 5 分でわかる！ OmegaT の使い方：初心者向けの短いチュートリアルですが、他の翻訳支援ツール経験者の方にも、おすすめです。新しいプロジェクトの作成から翻訳完了まで、一通りの流れを見ることができます。

- OmegaT のインストールと実行：この章は、OmegaT を最初に使いはじめるときに役立つでしょう。OmegaT のインストールと Windows、Mac OS X、Linux 各 OS 上での実行について、詳しい手順がここにあります。また上級者向けの話題として、コマンドラインモードとその可能性についても説明しています。

- ユーザーインターフェース、メニューとキーボードショートカット：この 2 つの章は、OmegaT のユーザーインターフェースおよび、メニューとキーボードショートカットから利用できる機能について、説明しています。おそらく頻繁に参照されるでしょう。

- プロジェクトの設定、OmegaT のファイルとフォルダー：OmegaT にとってプロジェクトとは、コンピューター翻訳支援ツールとして OmegaT が取り扱う作業単位を指します。最初の章では、たとえば原文と訳文の言語設定のような、プロジェクトの設定について説明しています。もうひとつの章では、翻訳プロジェクトが含むさまざまなフォルダーやファイルとその役割について取り上げています。また OmegaT に関連付けられた個人設定ファイルとプログラムファイルについても、説明しています。

- 翻訳入力行の動作：未翻訳の分節に関する翻訳入力行の動作設定をどのように行うか説明した、短い章です。
- プレーンテキストを扱うと整形されたテキストを扱う：この 2 つの章は、翻訳するテキストに関するいくつかの重要な点について説明しています。たとえば、文字エンコーディング（プレーンテキストファイルの場合）や、タグの取り扱い（整形されたテキストの場合）などです。
- 翻訳メモリ：翻訳メモリを配置できるさまざまなフォルダーの役割について説明するとともに、翻訳メモリに関連する多彩な情報を提供します。
- 原文の分節化：翻訳メモリツールは、分節と呼ばれるテキスト単位で処理を行います。OmegaT の分節は、段落単位か、あるいは分節化規則に基づく文単位です。段落単位の分節化はあまり使いませんが、いわゆる「文学的」なテキストの場合は役に立ちます。規則に基づく分節化は、たいていの場合、文単位の分節化を意味しています。分節化規則集には最初から多くの規則が定義されていますが、この章で説明される通り、ユーザーが独自に規則を追加することもできます。
- 検索と正規表現：OmegaT の検索は「『カンガルー』という単語を含む分節を一覧表示」といった感じで、簡単に使えます。その一方で、複雑な条件にも対応しています。たとえば、 2 個以上の連続するスペース文字を含む分節を検索したい、としましょう。この場合、「\s\s+」という正規表現パターンを使うことによって、問題の分節を検索することができます。正規表現は、分節化規則でも頻繁に使われています。
- 用語集、辞書、機械翻訳、綴り確認、LanguageTool：OmegaT　では辞書と用語集を幅広く活用できます。インターネットに接続していれば、Google Translate や Microsoft Translator のような機械翻訳サービスを　OmegaT　の中から利用できます。綴り確認機能を使うと、誤字脱字を検知し、翻訳中に修正することができます。オープンソースのツール LanguageTool は、文法や文体のありがちなミスを教えてくれます。
- その他：「データを失わないために」など、その他の興味深い話題を扱います。
- 付録は以下の情報を含みます
  - ウェブ上の OmegaT 情報：OmegaT のオンライン リソースに関する情報
  - 使用言語と言語コード：言語名と言語コードの ISO コード表
  - 編集ウィンドウのキーボードショートカット：編集ウィンドウで使えるショートカットの一覧
  - ショートカットのカスタマイズ：ショートカットをあなた好みにカスタマイズしましょう
  - トークナイザーとスクリプトの導入
  - チームプロジェクト
  - 法律上の表示と謝辞
- キーワード索引：関連情報を探しやすいように、充実したキーワード索引つきです。

第2章 5 分でわかる！ OmegaT の使い方

1. 新しいプロジェクトを作成する

注意：Mac ユーザーは、Ctrl キーを Command キー（⌘ マークが付いているキー）に読み替えてください

まずプロジェクトを作成しましょう。プロジェクトには、原文ファイルをはじめ、翻訳メモリや用語集、そしていずれ生成するはずの訳文ファイルなど、すべてのファイルが保持されることになります。プロジェクトメニューから新規作成...を選び、作成するプロジェクトの名前を入力してください。後で必要になるので、プロジェクトをどの場所に作成したか覚えておきましょう。

プロジェクト名を入力して保存したら、プロジェクトの新規作成ダイアログが開きます。ダイアログの先頭で、原文ファイルの言語と訳文ファイルの言語を選び、［確定］ボタンをクリックします。

他の設定に用があれば、Ctrl+E キーを押して、いつでもこのダイアログに戻ることができます。

続いて、翻訳対象ファイル一覧ダイアログが開きます。［原文ファイルを追加...］ボタンをクリックし、原文ファイルを選択します。OmegaT は選択されたファイルを、新しく作成されたプロジェクトの source フォルダーへコピーします。原文ファイルが編集ウィンドウに読み込まれたら、翻訳対象ファイル一覧ダイアログを閉じましょう。

2. ファイルを翻訳する

OmegaT は翻訳のために、原文を「分節」という単位に分割して表示します。ひとつの分節を翻訳し終えたら、Ctrl+U キーを押して、次の未翻訳分節へと移動します（Ctrl+Shift+U キーを押すと、次の翻訳済み分節へ移動します）。翻訳の成果を確認したくなったら、Ctrl+D キーを押して、いつでも訳文ファイルを生成できます。訳文ファイルは、プロジェクトの target フォルダー内に作成されます。翻訳作業のあいだ、［編集］メニューおよび［移動］メニューから、さまざまな役に立つ機能を利用することができます。

3. タグを検証する

もし原文ファイルが、Microsoft Word や LibreOffice Writer、HTML などの書式情報を持つファイル形式だった場合、OmegaT はその情報をタグに変換し、対象テキストをタグで括った形式で表示します。また文書には、書式とはまったく関係ないけれど、原文ファイル（と訳文ファイル）にとってやはり重要なタグが含まれることがあります。原文は次のように見えるはずです。

OmegaT は、**熱心な**翻訳者のための
*使いやすい*プログラムです。

一方 OmegaT は、この文を次のような形で表示します：

OmegaT は、<t0/>熱心な<t1/>翻訳者のための
<t2/>使いやすい<t3/>プログラムです。

OmegaT はタグを灰色で描画するので、見分けるのは簡単です。タグは保護されます。したがってタグの中身を書き換えることはできません。しかしタグごと削除したり、手入力したり、移動したりすることはできます。ただし、もしタグを誤って編集すると、訳文ファイルを開くときに問題が起きるかもしれません。そうならないよう、訳文ファイルを生成する前に Ctrl+Shift+V キーを押して、タグが正しいか検証しておくとよいでしょう。

4. 訳文ファイルを生成する

翻訳にタグのエラーが無いことを確認したら、Ctrl+D キーを押して訳文ファイルを生成しましょう。訳文ファイルは、プロジェクトの target フォルダー内に作成されます。

5. 覚えておきたい使い方のヒント

- ファイルが編集ウィンドウに読み込まれない場合は、OmegaT が原文ファイルのファイル形式に対応していないのかもしれません。「ファイルフィルター」の、OmegaT が対応しているファイル形式の一覧を確認してみてください。
- タスクに応じて、それぞれ個別に新規プロジェクトを作成しても構いませんし、原文ファイルはいつでも追加できます。
- プロジェクトの初期設定を確認したければ、Ctrl+E キーを押してプロジェクト設定ダイアログを開いてください。プロジェクトに含まれる原文ファイルの一覧を確認したければ、Ctrl+L キーを押して翻訳対象ファイル一覧ダイアログを開いてください。
- 翻訳作業の最後に、OmegaT は level1、level2、omegat と呼ばれる 3 つの翻訳メモリをプロジェクトフォルダー内に出力します。level1 と level2 は、他の翻訳支援ツールと翻訳メモリを共有するために使います。omegat は OmegaT 専用の翻訳メモリで、将来作成する別のプロジェクトで流用することができます。翻訳メモリファイルをプロジェクトの tm フォルダー内に配置しておくと、OmegaT はそこから「参考訳文」と呼ばれる類似分節を自動的に検索するようになります。
- Ctrl+Shift+G キーを押すと、新しい用語を追加することができます。また既存の用語集ファイルをプロジェクトの glossary フォルダーにコピーしておくと、OmegaT はそこから用語を自動的に検索するようになります。
- 原文や訳文から語句を検索したいときは、Ctrl+F キーを押せば、いつでもプロジェクトの検索ウィンドウを表示できます。
- もっと総合的な手引きをお探しの場合は、OmegaT のウェブサイトにある OmegaT for beginners [http://www.omegat.org/en/tutorial/OmegaT%20for%20Beginners.pdf]（英語）を参照してください。OmegaT に関するどんなことでも、困った場合は、OmegaT ユーザーグループ [http://tech.groups.yahoo.com/group/OmegaT/]に気軽に参加してください（日本語可）。

第3章 OmegaT のインストールと実行

1. Windows ユーザー

1.1. パッケージのダウンロード

Oracle の Java 1.6 JRE 互換の Java を実行可能な環境ですか？

- はい：OmegaT_3.n.n_Windows_without_JRE.exe をダウンロードしてください。
- いいえ / わかりません： OmegaT_3.n.n_Windows.exe をダウンロードしてください。

  このパッケージには Oracle の Java 実行環境（JRE）が含まれています。この JRE は、その環境にインストールされている可能性がある他の Java には影響しません。

1.2. OmegaT をインストールする

OmegaT をインストールするには、ダウンロードしたファイルをダブルクリックしてください

インストールの冒頭で、インストール中に使用する言語を選ぶことができます。続く画面で、インストール中に使用する言語を OmegaT 自体でも使うかどうか設定できます。該当するチェックボックスにチェックを入れた場合は、選択した言語を使うように OmegaT.l4J.ini ファイルをカスタマイズします（詳細は次の項を参照してください）。ライセンスに同意した後、セットアッププログラムでスタートメニューにフォルダーを作成するかどうかと、デスクトップとクイック起動バーに OmegaT アイコンを追加するかを指定できます。OmegaT.exe をデスクトップやスタートメニューへ右ボタンでドラッグすることにより、ショートカットを別途、作成できます。インストールの最後の画面では、リードミー（Readme）ファイルと、インストールしたバージョンの変更点を記載したファイルを見るかどうかを、選ぶことができます。

1.3. OmegaT の実行

いったん OmegaT をインストールしたならば、OmegaT.jar をクリックして直接起動することができます。また、コマンドラインから起動することもできます。

しかし、一番簡単なのはプログラム OmegaT.exe を実行することです。この場合の起動時の引数は、OmegaT.exe ファイルと同じ場所にある OmegaT.l4J.ini ファイルから読み込まれます。このファイルを編集することで起動引数を変更できます。以下に示した INI ファイルは、最大で 1GB のメモリを確保し、ユーザー言語として日本語を、国に日本を指定した例です：

```
# OmegaT.exe runtime configuration
# To use a parameter, remove the '#' before the '-'
# Memory
-Xmx1024M
# Language
-Duser.language=JA
# Country
-Duser.country=JP
```

参考：Windows のリモートデスクトップ経由で OmegaT を使っていて動作が遅い場合は、次のオプションを試してみてください。

```
-Dsun.java2d.noddraw=false
```

1.4. OmegaT のアップグレード

この情報は「通常の」Windows 版 OmegaT のみを対象としています。Java Web Start 版（自動更新されます）や、Windows 上にインストールされたクロスプラットフォーム版は対象外です。

OmegaT をすでにインストール済みで、それをより新しいバージョンにアップグレードしたい場合は、２つの選択肢があります。

- 上書きインストール：新しいバージョンをインストールする際、単純にインストール先として既存の OmegaT と同じフォルダーを指定します。「古い」バージョンの OmegaT は上書きされますが、個人設定は引き継がれます。この中には、OmegaT の各種設定、 OmegaT.l4J.ini ファイルに対する変更、そして（もしあれば）独自に作成した起動スクリプト（バッチファイル）が含まれます。

この方法では、「Windows without JRE」版の方をダウンロードしても構いません。その場合、新しいバージョンは実行時に既存の JRE を使います。

- 新しいフォルダーへインストール：この方法を使うと、古いバージョンと新しいバージョンを共存させることができます。新しいバージョンに慣れるまでは、古いバージョンも使い続けたいという場合に、適しています。この方法でも、OmegaT の設定を引き継ぐことができます。ただし、次のような制限があります：
  - OmegaT.l4J.ini ファイルに変更を加えていたり、バッチファイルを使っている場合は、それらを新しいフォルダーへコピーする必要があります。
  - 既存の OmegaT が「Windows with JRE」バージョンの場合は、新しいバージョンも「Windows with JRE」でなければなりません。

2. Linux（Intel）ユーザー

2.1. 適切なパッケージのダウンロード

Oracle の Java 1.6 JRE 互換の Java を実行可能な環境ですか？

- はい：OmegaT_3.n.n_Without_JRE.zip をダウンロードしてください。
- いいえ / わかりません：OmegaT_3.n.n_Linux.tar.bz2 をダウンロードしてください。

  このパッケージには Oracle の Java 実行環境（JRE）が含まれています。この JRE は、その環境にインストールされている可能性がある他の Java には影響しません。

2.2. OmegaT をインストールする

ダウンロードしたファイルを展開（unpack/untar）してください。作業フォルダーに omegat フォルダーが作成され、OmegaT の実行に必要なすべてのファイルが作成されます。.tar.gz ファイルを展開するコマンドは以下の通りです：

$ tar xf downloaded_file.tar.gz

2.3. メニュー（KDE）またはパネル（GNOME）への OmegaT の追加

2.3.1. KDE 4 ユーザー

以下の手順で OmegaT をメニューへ追加できます：

- Alt+F2 を押して KRunner を表示します。kmenuedit+enter と入力し、コマンドを実行します。KMenuEditor が表示されます。KMenuEditor から［ファイル］→［新規アイテム］を選びます。
- 追加したいメニューを選択し、［ファイル］→［新規サブメニューとファイル］→［新規アイテム］を選択します。［新規アイテム名］として OmegaT と入力します。
- ［コマンド］欄には、ナビゲーションボタンを使用して OmegaT 起動スクリプト（展開したフォルダーの中にある OmegaT という名称のファイル）を指定します。

- ［アイコン］ボタン（［Name/Description/Comment］欄の右にあります）をクリックします。
- ［その他のアイコン］→［閲覧］を押し、OmegaT プログラムフォルダーにある images フォルダーを選択します。アイコンとして OmegaT.png を選択します。
- 最後に［ファイル］→［保存］ で変更を保存します。

2.3.2. GNOME ユーザー

以下の手順で OmegaT をメニューへ追加できます：

- パネルを右クリックし、［新規ランチャーを追加］を選択します。
- ［名前］欄に OmegaT と入力します。［コマンド］欄には、ナビゲーションボタンを使用して OmegaT 起動スクリプト（展開したフォルダの中にある OmegaT という名称のファイル）を指定します。最後に［OK］をクリックします。
- ［アイコン］ボタンを押し、［閲覧...］を押し、OmegaT のプログラムフォルダーにある images フォルダーを選択します。アイコンとして OmegaT.png を選択します。GNOME は、何も指定しないとアイコンに SVG ファイルを想定するらしく、ファイル形式に対応しているはずのアイコンファイルを表示できないことがあります。しかし、上記のようにフォルダーを選択すれば、OmegaT.png を選択できるようになります。

2.4. OmegaT の実行

OmegaT の実行は、起動オプションを記述したスクリプトを使ってコマンドラインから可能です。あるいは OmegaT.jar をクリックして直接起動することもできます。その方法はディストリビューションによって異なります。PATH の設定が正しいことと、.jar ファイルが正しく Java に関連づけられていることを確認してください。詳細は「コマンドラインから OmegaT を起動する」を参照してください。

3. Mac OS X ユーザー

3.1. パッケージのダウンロード

OmegaT には Java JRE 1.7 が含まれています。

OmegaT_3.n.n_Mac.zip をダウンロードしてください。

3.2. OmegaT をインストールする

OmegaT_3.n.n_Mac.zip をダブルクリックして解凍してください。OmegaT フォルダーが作成されます。フォルダーには 2 つのファイルが含まれています。index.html と OmegaT.app です。このフォルダーを適切な場所（アプリケーションフォルダなど）へコピーしてください。ここまでくれば、OmegaT_3.n.n_Mac.zip ファイルを削除してもかまいません。

3.3. OmegaT のドックへの追加

OmegaT.app ファイルをドックへドラッグ＆ドロップします。

3.4. OmegaT の実行

OmegaT.app をダブルクリックするか、ドックのアイコンをクリックします。

OmegaT の挙動は、パッケージに含まれる Properties と OmegaT.sh ファイルを編集することでカスタマイズできます。

まず OmegaT.app を右クリックして、［パッケージの内容を表示］を選びます。次に、Contents/MacOS フォルダーにある OmegaT.sh を右クリックして、任意のテキストエディターで開きます。ある

いはコマンドラインから「cd」コマンドで目的の階層に移動し、emacs や vi のようなコマンドラインエディターで OmegaT.sh を開いてもよいでしょう。

OmegaT.sh を編集することで、設定を変更できます。定義済みの設定を有効にするには、引数の先頭の # 記号を削除してください。たとえば、LANGUAGE="-Duser.language=ja"（# 記号なし）とすると、OmegaT は日本語のユーザーインターフェースで起動します。

メモリ容量の割り当てを変更する場合は、起動行を直接編集します。たとえば、${JAVA} -Xmx2048m ${MACOS} ${LANGUAGE} ${COUNTRY} ${PROXY_HOST} ${PROXY_PORT} ${GOOGLE_API_KEY} ${MS_CLIENT_ID} ${MS_CLIENT_SECRET} ${MY_MEMORY_EMAIL} ${TAAS_USER_KEY} -jar OmegaT.jar とすると、OmegaT は 2 ギガのメモリ割り当てで起動します。

OmegaT.app を複数起動したい場合は、OmegaT.app/Contents/MacOS/ にある OmegaT.sh ファイルをダブルクリックしてください。

OmegaT をコマンドラインから起動する場合は、OmegaT.app/Contents/MacOS/Java/ にある OmegaT.jar ファイルを指定してください。詳細は「コマンドラインから OmegaT を起動する」を参照してください。

3.5. Max OS X での利点

OmegaT.app は Mac OS X の「サービス」を利用できます。Spotlight や Google などをサービスとして登録しておくと、OmegaT で選択した任意の単語をサービスで検索することができます。また AppleScript や Automator を使用して、よく行う作業をサービスやスクリプトとして登録しておくこともできます。

4. その他のシステム

この情報は、Solaris SPARC/x86/x64、Linux x64/PowerPC、Windows x64 などのシステムを対象としています。

4.1. 適切なパッケージのダウンロード

OmegaT は Linux（Intel x86）と Windows プラットフォーム向けにバンドルされた Oracle Java JRE で動作します。その他のプラットフォーム（Linux PowerPC、Linux x64、Solaris SPARC/x86/x64、Windows x64 など）を使用しているユーザーは、互換性のある Java JRE を動作できる環境であれば OmegaT を使用できます。

Oracle の Java 1.6 JRE 互換の Java を実行可能な環境ですか？

- はい：OmegaT_3.n.n_Windows_without_JRE.zip をダウンロードしてください。Java 1.6 JRE 互換の JRE がインストールされていれば、どのプラットフォームであっても、このパッケージが使用できます。

- わからない： ターミナルウィンドウを開いて、コマンド「java -version」を実行してください。「command not found」のようなメッセージが表示された場合は、そのシステムには Java がインストールされていない可能性があります。

- いいえ：システムに適した Java JRE（下記参照）を入手してください。その後 OmegaT_3.n.n_Without_JRE.zip をダウンロードしてください。

  Oracle は以下のシステム向けの JRE を提供しています：Solaris SPARC/x86（Java 1.6）と Linux x64、Solaris x64、Windows x64（Java 1.6）http://www.oracle.com/technetwork/java/archive-139210.html

  IBM は Linux PowerPC 向けに JRE を提供しています：http://www.ibm.com/developerworks/java/jdk/linux/download.htm [http://www.ibm.com/developerworks/java/jdk/linux/download.html]

  必要なパッケージ各々のインストール指示に従ってください。

4.2. OmegaT をインストールする

OmegaT をインストールするには、OmegaT_3.n.n_Without_JRE.zip ファイルを展開してください。作業フォルダーに OmegaT_3.n.n_Without_JRE フォルダーと、OmegaT の実行に必要なすべてのファイルが作成されます。

4.3. 起動に便利なショートカットの作成

便利な場所に OmegaT のショートカットを作成するには、それぞれのシステムのマニュアルを参照してください。

4.4. OmegaT の実行

OmegaT をインストールした後は、コマンドラインからの直接実行や、起動引数を記述したスクリプトによる実行も可能です。あるいは OmegaT.jar をクリックして直接起動することもできます。その方法はディストリビューションによって異なります。PATHの設定が正しいことと、.jar ファイルが正しく Java に関連づけられていることを確認してください。詳細は「コマンドラインから OmegaT を起動する」を参照してください。

5. Java Web Start の使用

Java Web Start テクノロジー（Java 1.6 以降が対応）を用いると、独立型の Java アプリケーションを、１クリックでネットワーク上からローカル環境に配置することができます。Java Web Start を使うと、適切なバージョンの Java 実行環境（JRE）と、最新バージョンのプログラムとを確実に配布できるという利点があります。Java Web Start を使用して OmegaT を最初に起動する場合は、ブラウザ上で以下の URL にアクセスしてください：

http://omegat.sourceforge.net/webstart/OmegaT.jnlp

OmegaT.jnlp ファイルをダウンロードし、それをクリックしてください。インストール中、OS によってはいくつかのセキュリティ警告が表示される場合があります。この（Java Web Start 版）実行ファイルに与える権限（「コンピューターへの無制限アクセス」と表示される可能性があります）は、通常のローカル版に対して適用するものと同一で、コンピューターのハードディスクにアクセスすることを許可するものです。インストール後は、OmegaT.jnlp をクリックすると、プログラムの更新がないか検索し、もしあればインストールした後、OmegaT を起動します。最初のインストールが終わっていれば、オフラインであっても OmegaT.jnlp を使用できます。

プライバシーについて：OmegaT Java Web Start は、実行中のコンピューターに関するいかなる情報も外部へ保存することはありません。プログラムは導入されたコンピューター上のみで実行されます。作成した文書や翻訳メモリは使用しているコンピューターに保存され、OmegaT 開発プロジェクトがその作業内容やその他の情報へアクセスすることはありません。

もし、起動時に何らかの引数を指定したい場合（上記参照）は、通常のインストールを行う必要がある点に注意してください。

6. コマンドラインから OmegaT を起動する

通常は、コマンドラインから OmegaT を起動する必要はありません。しかしコマンドライン起動からだと、ユーザーの好みに応じてプログラムを制御し、その挙動を変えることができます。コマンドラインを使用した OmegaT の起動方法は、大きく２つあります。

6.1. コマンドラインウィンドウを開く

コマンドラインウィンドウは「ターミナルウィンドウ」とも呼ばれます。Windows では「コマンドプロンプト（または MS-DOS プロンプト）」と呼ばれ、［スタートメニュー］→［すべてのプログラム］→［アクセサリ］→［コマンドプロンプト］と選択して開くことができます。Mac OS X の場合は、［アプリケーション］→［ユーティリティ］フォルダにある「ターミナル］プログラムがこれにあたります。

OmegaT の起動には、通常は 2 つのコマンド実行が必要です。まず最初に：

cd （フォルダー）

（フォルダー）には、OmegaT プログラムのうち OmegaT.jar がある場所のフォルダー名を、絶対パスで指定します。たとえば、以下のように実行します：

Windows の場合

cd C:\Program Files\OmegaT

MAC OS X の場合

cd <OmegaT.app location>/OmegaT.app/Contents/Resources/Java/

Linux の場合

cd /usr/local/omegat

このコマンドで、カレントフォルダーを、OmegaT 実行ファイルがあるフォルダーに移動します。2 番目のコマンドで実際に OmegaT を起動します。もっとも基本的な形式では、このコマンドは次のようになります：

java -jar OmegaT.jar

大文字、小文字の区別に注意してください。Windows 以外の OS では、OmegaT ではなく omegat と入力すると、起動しません。

何か問題が発生した場合は、その原因を調べる上で、この起動方法が役に立ちます。プログラム使用中に何らかのエラーが発生した場合、原因調査の助けになる情報がエラーメッセージとしてターミナルウィンドウに出力されます。

上記の起動方法は、日常的に行うプログラム実行の手順としては、あまり実用的ではありません。そこで、2 つのコマンドをファイル（「スクリプト」または Windows 上では「.bat ファイル（バッチファイル）」と呼ばれます）に記述しておきます。

このファイルを実行すると、記述した複数のコマンドを自動的に処理できます。その結果、起動コマンドを変更したい場合は、ファイル内容の修正を行うだけでよいことになります。

6.2. 起動コマンドの引数

基本的なコマンドはすでに上記で述べた通りです。このコマンドを変更するには「引数」を追加します。引数を追加する位置は、最初の「java」の直後、「-jar OmegaT.jar」の直前です。Windows では、OmegaT.l4J.ini を書き換えることによって、好みの設定に変更できます。それ以外のプラットフォームでは、お使いのランチャー（たとえば Mac では OmegaT.sh、Linux では OmegaT）を修正することによって同様に変更できます。

指定できる引数の一覧を以下に示します。上級者のユーザーは、ターミナルウィンドウ上で man java と入力すると、引数の詳細情報を得られます。

- ユーザーインターフェース言語

  -Duser.language=XX 通常、何も引数を指定せずに OmegaT を起動した場合は、プログラムはまず OS の言語を検出します。OmegaT のユーザーインターフェースがその言語に対応していれば、OmegaT はそれを使用します。したがって、たとえば OS がロシア語であれば、OmegaT のユーザーインターフェースやメニューなどはロシア語で表示されます。もし OS の言語に対応していない場合、OmegaT はデフォルトの英語で表示されます。これが基本的な仕組みです。

  「-Duser.language=XX」引数の指定により、OmegaT は OS の言語よりも優先して、指定した言語を使用するようになります。引数の中の「XX」は、指定したい言語の 2 桁の言語コードです。OmegaT を（たとえばロシア語 OS 上で）日本語のユーザーインターフェースで使用したい場合は、実行コマンドは以下のようになります：

java -Duser.language=ja -jar OmegaT.jar

- ユーザーの地域設定

  -Duser.country=XX 言語と合わせて、国を指定することもできます。例えば日本の場合、JP を指定します。特定の言語版の「5 分でわかる！ OmegaT の使い方」を表示するには、言語と国の両方を指定する必要があります。「ポルトガル語 - ブラジル」の組み合わせに対する pt_BR のように、 1 つの言語に 1 つの国のみの組み合わせである場合であっても、これは必要です。

- メモリの割り当て

  -XmxZZM この引数により、OmegaT により多くのメモリを割り当てられます。デフォルトでは 512 MB が割り当てられます。そのため、これより小さな値をこの引数で指定しても効果はありません。「ZZ」には、割り当てたいメモリ容量を MB（メガバイト）単位で指定します。1024MB（1GB）のメモリを割り当てて OmegaT を起動する場合のコマンドは：

  java -Xmx1024M -jar OmegaT.jar

- プロキシサーバーのホスト名と IP アドレス

  -Dhttp.proxyHost=nnn.nnn.nnn.nnn プロキシサーバーの IP アドレス。

- プロキシサーバーのポート番号

  -Dhttp.proxyPort=NNNN プロキシサーバーのポート番号。

- Google Translate v2

  -Dgoogle.api.key=A123456789B123456789C123456789D12345678 Google Translate サービスのプライベートな Google API キー。キーの文字列長は 38 文字であることに注意してください。

- Microsoft Translator

  事前に無料の Microsoft アカウントを取得しておいてください。それには Windows Azure Marketplace [http://datamarket.azure.com/dataset/bing/microsofttranslator#schema] に登録し、Translator サービスを使用（サブスクライブ）する必要があります。一ヶ月 200 万字までは、無料で使えます。アカウント情報 [https://datamarket.azure.com/account]のページにある 2 つの情報、プライマリ アカウント キーと顧客 ID が必要になります。

  -Dmicrosoft.api.client_id=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

  -Dmicrosoft.api.client_secret=XXXX9xXxX9xXXxxXXX9xxX99xXXXX9xx9XXxXxXXXXX=

複数の引数を、組み合わせて指定することもできます。上記で例示したすべての引数を使用して OmegaT を起動する場合、コマンドは以下のようになります：

java -Dswing.aatext=true -Duser.language=pt -Duser.country=BR -Xmx1024M -Dhttp.proxyHost=192.168.1.1 -Dhttp.proxyport=3128 -jar -OmegaT.jar

6.3. OmegaT をコマンドラインモードで使う

コンソールモードは、スクリプト環境で OmegaT を翻訳ツールとして用いる場合に使用できます。コンソールモードで開始すると、どのコンソール上でも動作するように、GUI を表示しません。そして、指定されたプロジェクトを自動的に処理します。

6.3.1. 必要な条件

OmegaT をコマンドラインモードで実行するためには、有効な OmegaT プロジェクトが存在している必要があります。プロジェクトの場所は、起動時にコマンドライン上で指定するので、どこであってもかまいません。

別の設定を用いる場合は、有効な設定ファイルも存在している必要があります。その場合は、次の 2 つのうちいずれかの方法をとってください：

- OmegaT を通常通り（GUI 経由で）起動して、設定しておきます。そのままコンソールモードで起動すると、同じ設定を使用します。
- OmegaT を通常通り起動できない（GUI 環境がない）場合は、次の方法をとってください：同じ端末の他の場所、または別の端末にある OmegaT の設定ファイルを、あらかじめ決めたフォルダーにコピーしておきます。そのフォルダーの場所は、起動時にコマンドライン上で指定するので、どこであってもかまいません。設定ファイルは、filters.conf と segmentation.conf です。ユーザーのホームフォルダー（Windows の場合 C:\Documents and Settings\%User%\OmegaT、Linux の場合 %user%/.omegat/）にあります。

6.3.2. コンソールモードで起動する

コンソールモードで OmegaT を起動する場合、いくつかの引数が必要になります。最も重要な引数は、<project-dir> です。他に任意の引数として、 --config-dir=<config-dir>があります。例：

java -jar OmegaT.jar /path/to/project \

--config-dir=/path/to/config-files/ \

--mode=console-translate|console-createpseudotranslatetmx|console-align

--source-pattern={正規表現}

すべての引数が、二重のハイフン記号（-）で始まることに注意してください。

説明：

- <project-dir> プロジェクトがどの場所にあるかを指定します。この引数を指定すると、OmegaT がコンソールモードで起動し、指定されたプロジェクトを翻訳します。
- --config-dir=<config-dir> 設定ファイルがどのフォルダーにあるかを指定します。このオプションが指定されない場合は、OmegaT は初期設定を使用します（ユーザーホームフォルダーの下の OmegaT フォルダーか、それが存在しない場合は現在の作業フォルダー）。引数は二重のハイフン記号（-）で始まることに注意してください。
- --mode=...- この引数を指定すると、OmegaT がコンソールモードで起動し、以下に示すサービスを自動的に実行します。
  - --mode=console-translate

    このモードでは、source フォルダーのファイルを翻訳メモリを使って翻訳します。OmegaT が TMX ファイルと共にサーバーで実行され、プロジェクトを自動処理するような状況に向いています。
  - --mode=console-createpseudotranslatetmx

    このモードでは、原文ファイルのみを元にして、プロジェクト全体を対象にした翻訳メモリを生成します。TMX ファイルに関する設定は、次の引数で指定します：

    --pseudotranslatetmx=allsegments.tmx --pseudotranslatetype=[equal|empty]

    引数 pseudotranslatetype を指定すると、原文と同じ対訳になるか、あるいは空になります。
  - --mode=console-align

    このモードでは、source フォルダーにある Java の properties ファイルを整列します。整列した TMX ファイルは、align.tmx という名前で omegat フォルダーに出力されます。

    突き合わせる訳文ファイルの場所は、下記の引数で指定します。

    --alignDir={訳文ファイルの場所}

    alignDir には、プロジェクトの訳文言語に翻訳された訳文ファイルを配置しておく必要があります。たとえばプロジェクトが英→日翻訳の言語設定になっている場合、alignDir フォルダーには、

ファイル名が「_ja」で終わるバンドルが存在しなければなりません。整列した TMX ファイルは、align.tmx という名前で omegat フォルダーに出力されます。

- --source-pattern={正規表現}

  この引数を指定すると、どのファイルを自動処理するのか、絞り込むことができます。引数を省略した場合は、すべてのファイルが処理されます。以下に、典型的な指定例をいくつか挙げます：

  - .*\.html

    すべての HTML ファイルを処理します。正規表現の規則にしたがって、拡張子の区切りを示すドット「.」は「\.」のようにエスケープする必要があります。

  - test\.html

    source フォルダーにある「test.html」というファイルのみを処理します。他のフォルダーに「test.html」という同名のファイルがあったとしても、無視されます。

  - dir-10\\test\.html

    dir-10 フォルダーにある「test.html」というファイルのみを処理します。パス区切り文字の「\」は、やはり「\\」のようにエスケープする必要があります。

- --output-tag-validation-={正規表現}

  この引数を指定すると、どのファイルを自動処理するのか、絞り込むことができます。引数を省略した場合は、すべてのファイルが処理されます。以下に、典型的な指定例をいくつか挙げます：

  - .*\.html

    すべての HTML ファイルを処理します。正規表現の規則にしたがって、拡張子の区切りを示すドット「.」は「\.」のようにエスケープする必要があります。

  - test\.html

    source フォルダーにある「test.html」というファイルのみを処理します。他のフォルダーに「test.html」という同名のファイルがあったとしても、無視されます。

  - dir-10\\test\.html

    dir-10 フォルダーにある「test.html」というファイルのみを処理します。パス区切り文字の「\」は、やはり「\\」のようにエスケープする必要があります。

- --tag-validation=[abort|warn] 出力ファイル名

  この設定は、バッチモードでタグ検証を可能にします。「abort」を指定した場合は、不正なタグを見つけた時点で、タグ検証を中止します。「warn」を指定した場合は、すべての分節についてタグ検証を実施し、不正なタグについては警告を出力します。

- --no-team チーム作業用のプロジェクトに関連する引数です。この引数を指定すると、チームプロジェクトを開いても同期しません。

- --disable-project-locking Windows において、OmegaT の複数のインスタンスが同じプロジェクトを開けるようになります。Windows 上では、OmegaT がプロジェクトを開くと、既定で omegat.project ファイルをロックします。そのため、OmegaT の別のインスタンスが使用中のプロジェクトを開こうとすると、エラーメッセージが表示されます。この引数を指定すると、ロックしません。

6.3.3. サイレントオプション

追加のコンソールモードのオプションとして、--quiet があります。このオプションを指定すると、画面出力が極力抑えられます。通常ステータスバーに表示されるようなメッセージは、表示されなくなります。

使い方: java -jar OmegaT.jar /path/to/project --mode=console-translate --quiet

6.3.4. タグ検証の設定

もう 1 つ別の追加のコンソールモードのオプションとして、--tag-validation=[abort|warn] があります。このオプションを指定すると、翻訳や整列の処理時にタグ検証を行います。オプション値「abort」を指定した場合は、タグのエラーがあった場合はエラーを出力して処理を中止します。オプション値「warn」を指定した場合は、タグのエラーがあった場合はエラーを出力して処理を継続します。

使い方: java -jar OmegaT.jar /path/to/project --mode=console-translate --tag-validation=abort

7. ソースコードからの OmegaT のビルド

OmegaT を独自にビルドするためには、ant プログラム（http://ant.apache.org/bindownload.cgi）も必要になります。OmegaT_3.n.n_Source.zip ファイルを展開し、OmegaT_3.n.n_Source フォルダーに移動するか、あるいは SVN リポジトリからソースコードをチェックアウトし、omegat フォルダー（チェックアウト時に名前を変えた場合は何であれそのフォルダー）に移動してください。build.xml ファイルがそのフォルダーに存在することを確認してください。それから、以下のコマンドを入力してください：

```
$ ant jar release
```

これにより、dist フォルダーに OmegaT の配布版が生成され、OmegaT の実行に必要なすべてのファイルが揃います。

第4章 OmegaT のインターフェース

1. OmegaT のメインウィンドウ、その他のウィンドウ、ダイアログ

OmegaT のメインウィンドウは、メインメニュー、ステータスバーと、いくつかのウィンドウからなります。またプロジェクトの設定変更時に使用するような、追加のウィンドウやダイアログもあります。以下で、それらの用途と呼び出し方を簡単に説明します：

表4.1 OmegaT のメインウィンドウ

編集ウィンドウ	翻訳内容を入力、編集する場所です。
参考訳文ウィンドウ	翻訳メモリから、最も類似した分節を表示します。
用語集ウィンドウ	現在の分節で見つかった用語を表示します。
辞書ウィンドウ	現在の分節で見つかった辞書項目を表示します。
機械翻訳ウィンドウ	機械翻訳サービスによる翻訳を表示します。
メモウィンドウ	現在の分節に関するメモを表示または編集します。
コメントウィンドウ	PO ファイルのコメントや、翻訳中の属性値の名前（XHTML ファイルの場合）を表示します。

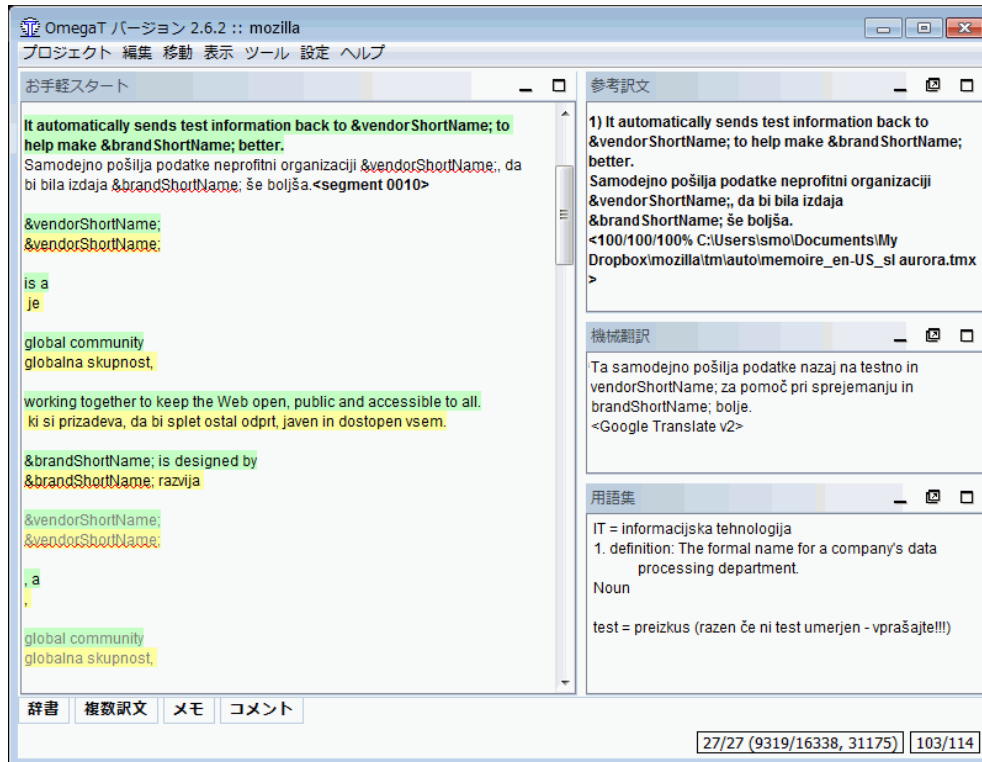
表4.2 その他のウィンドウ

タグ検証ウィンドウ	タグ検証の結果を閲覧できます。［Ctrl+Shift+V］で開き、［Esc］で閉じます。
ヘルプウィンドウ	取扱説明書を閲覧できます。［F1］で開き、［Esc］で閉じます。
翻訳状況ウィンドウ	プロジェクトの翻訳状況を表示します。［ツール］→［翻訳状況］からウィンドウを開きます。
参考訳文の一致率ウィンドウ	プロジェクトの参考訳文の一致率を閲覧できます。［ツール］→［参考訳文の一致率］からウィンドウを開きます。

表4.3 設定ダイアログ

プロジェクト設定	プロジェクトで使用する言語とフォルダー構成を変更できます。ショートカット［Ctrl+E］または［プロジェクト］→［プロジェクト設定...］で開き、［Esc］で閉じます。
字体設定	OmegaT が使用する字体を変更します。原文、訳文、参考訳文、用語集の表示に使用されます。メインメニューの［設定］→［字体...］で開き、［Esc］ で閉じます。
ファイルフィルター	OmegaT が対応するファイル形式の取り扱い方を設定します。［設定］→［ファイルフィルター...］で開き、［Esc］で閉じます。
分節化規則	文章を分節化する方法を設定します。［設定］→［分節化規則...］で開き、［Esc］で閉じます。
翻訳入力行	分節を移動する際の OmegaT の挙動を設定します。［設定］→［翻訳入力行...］で開き、［Esc］で閉じます。

2. OmegaT メインウィンドウ

図4.1 OmegaT メインウィンドウ

メインウィンドウは、いくつかのウィンドウ、メインメニュー、ステータスバーからなります。ウィンドウの名称部分をドラッグすると、表示位置を変更したり、独立したウィンドウとして表示させせることができます。各ウィンドウの右上角には、その表示状態によって異なる、以下のようなボタンが表示されます。

注記

本来、開いていたり最小化していたりするはずのウィンドウが見当たらない場合は、［設定］→［ウィンドウを初期値に復元］を選んでください。ウィンドウの配置をインストール直後の状態に復元します。

表4.4 ウィンドウ表示アイコン

	ウィンドウを最小化します。そのウィンドウの名称だけが、メインウィンドウの底部に表示されます。
	ウィンドウを最大化します。

	ウィンドウを最大化される前の状態に戻します。
	ウィンドウをメインウィンドウから切り離して、独立したウィンドウとして表示します。
	ウィンドウを、独立したウィンドウから、メインウィンドウ内の子ウィンドウ表示に戻します。

ウィンドウは重ねて表示させることもできます。この場合、ウィンドウの上部にタブが表示されます。ウィンドウ間のセパレーターをドラッグすると、ウィンドウの大きさを変更できます。ウィンドウの配置を変更しているうちに収拾がつかなくなった場合は、［設定］→［ウィンドウを初期値に復元］を選んでください。ウィンドウの配置がインストール直後の状態に復元されます。

メインウィンドウの右下角には、翻訳の進捗を表示します（下表左列の数字は、上図を参照しています）:

表4.5 メインウィンドウ - 進捗

27/27	現在のファイルの分節数：翻訳済み / すべて
9319/16338	繰り返しを除いたプロジェクトの分節数：翻訳済み / すべて
31175	プロジェクトの分節数（繰り返しを含む）
103/114	現在分節の文字数：原文 / 訳文

実用上の観点から言えば、最も重要な数値は 2 番目のペアです。これは翻訳がどこまで進んでいるかを示すもので、プロジェクトの分節数と関係しています。例示されたプロジェクトは、繰り返しを除いたプロジェクトの分節がすべて翻訳済みになっていることから明らかなように、翻訳が完了していることが分かります。

2.1. 編集ウィンドウ

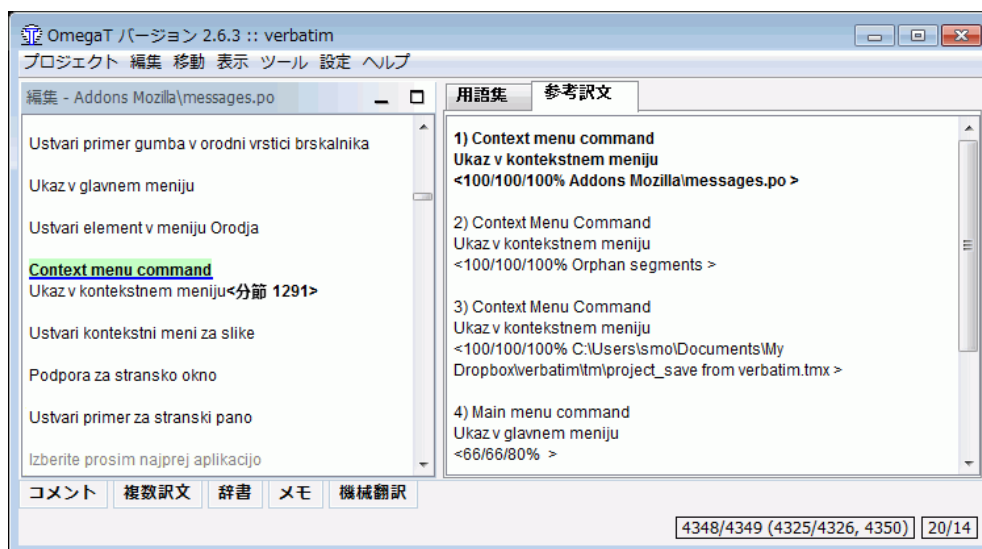
翻訳内容を入力、編集する場所です。編集ウィンドウには、部分的に翻訳されたテキストが表示されます。つまり、翻訳済みのテキストは訳文言語で表示され、未翻訳のテキストは原文言語で表示されます。テキストは分節の集まりとして分割表示され、スクロールで文書内を移動でき、どの分節でもダブルクリックすることで編集できます。上記の例では、翻訳済みの分節が黄色でハイライト表示されています（［表示］→［翻訳済み分節を色づけ］を有効に設定しているため）。

その分節のうちの 1 つは、現在分節です。現在分節を見ると、2 つの部分から構成されていることが分かります。上側には原文が緑色背景の太字で表示されます。下側が翻訳入力行で、最後は分節番号です。分節番号は <分節 nnnn> の形式で表示され、「nnnn」はプロジェクト中における分節の番号を表しています。上側の原文を参照しながら、下側の翻訳入力行の内容を修正または上書きし翻訳を進めてください。

翻訳入力行の設定により、未翻訳の分節の編集領域は、空欄になっているか、原文と同じテキスト、または最も一致率の高い参考訳文が入った状態になります。別の分節に移ると、入力した訳文が確定され、翻訳メモリに蓄積されます。その分節の訳文を原文と同じにしたいときは、編集領域のテキストを削除して空欄にしておいてください（［Ctrl+A］で全選択して［Del］で削除するか、右クリックから［訳文を削除］を選びます）OmegaT　は訳文が原文とまったく同じ状態でも記憶することができます。これは商標や固有名詞、第 3 の言語の一部など、翻訳の必要がない部分を残しておきたいときに便利です。詳細は、「翻訳入力行の設定」を参照してください。

編集ウィンドウ上で右クリックした場合はポップアップメニューが開き、［切り取り］、［コピー］、［貼り付け］（ショートカットの　［Ctrl+X］、［Ctrl+C］、［Ctrl+V］と同じ）および［分節へ移動］コマンドを利用できます。

2.2. 参考訳文ウィンドウ

図4.2 参考訳文ウィンドウ

参考訳文ウィンドウは、翻訳メモリの中から最も一致率の高い分節を表示します。検索対象となるのは、そのプロジェクトの翻訳中にリアルタイムで蓄積されていく内部翻訳メモリはもちろん、過去の翻訳作業で使用したものや、顧客や翻訳会社から受け取ったものをインポートして使用することもできます。

次の分節に移動すると、最初の参考訳文（最も一致率の高いもの）が自動的に選択されます。ここで［Ctrl+2］、［Ctrl+3］、［Ctrl+4］または［Ctrl+5］　を押して、別の参考訳文を選択することもできます。もちろん、5 番目の参考訳文が存在しないときに［Ctrl+5］を押しても、何も起こりません。選択した（太字で表示された）参考訳文を、現在編集中の訳文と置換して使用するにはCtrl+R を押します。現在のカーソル位置に挿入する場合は［Ctrl+I］を押します。

一致率は、簡単に言えば、比較する分節どうしで共通している単語数を、長い方の分節の単語数で割ったものです。選択された状態の参考訳文は太字で表示されます。翻訳しようとしている分節に存在しない単語は青字で、それに隣接する単語は緑字で表示されます。上記の例では、「Context menu command」が原文分節です。先頭の参考訳文は、一致率　100%　です。すべての単語が一致しています。続く 2 つの参考訳文も同じ一致率ですが、4 番目は似て非なるものです。一致率を表示している行には他にも、その参考訳文を収録している翻訳メモリの名前が表示されます。ファイル名が付記されていないものは、プロジェクトの内部翻訳メモリからの表示です。孤立した分節（上図での 2 番目の参考訳文）とは、そのプロジェクトの翻訳メモリ中に存在するが、原文の分節とは一致しなくなった分節を指します。

一致率には 3 種類の数値が表示されます（上図の 4 番目の参考訳文で言うと「66/66/80」の部分）。定義は、次の通りです：

- トークナイザーによる一致率
- デフォルトの OmegaT の一致率 - 数字とタグを除外して一致した単語数を、総単語数で割った数値

- 数字とタグを含む OmegaT の一致率

2.2.1. 参考訳文ウィンドウのカスタマイズ

［ 設定 ］→［ 外部 TMX ファイル... ］で表示されるダイアログから、参考訳文ウィンドウの表示方法を設定できます。

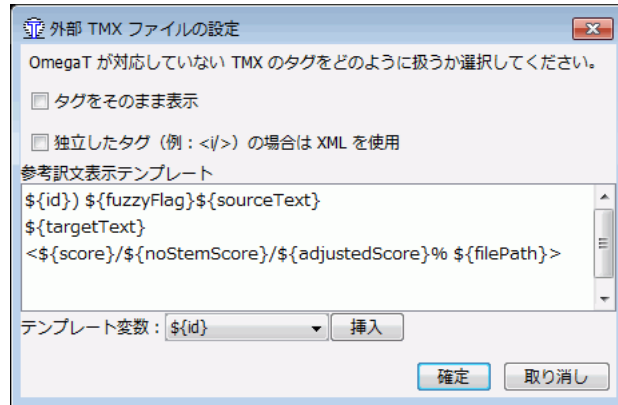
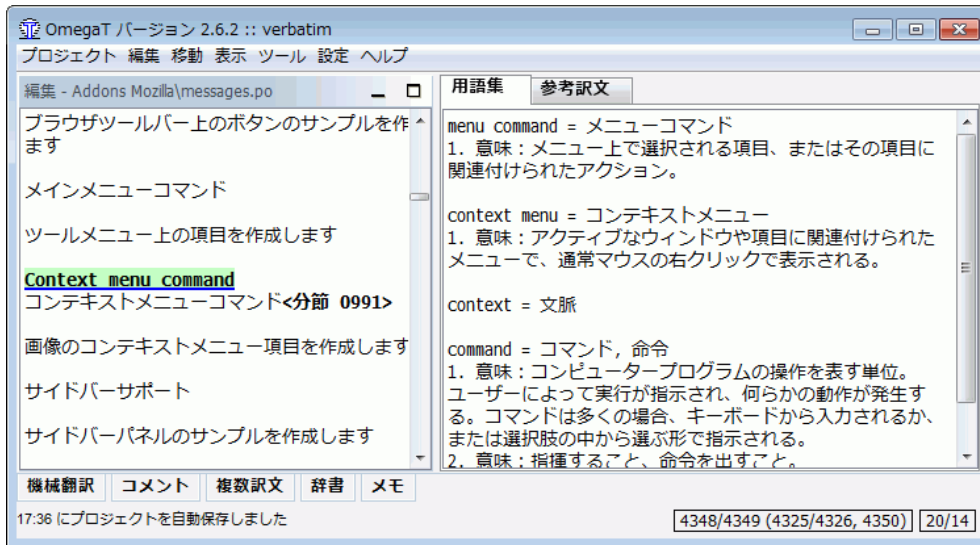
図4.3 参考訳文ウィンドウの設定

上の図は、参考訳文表示テンプレートのデフォルトの状態を示しています。表示内容は、次の変数を使ってカスタマイズできます。

表4.6 参考訳文ウィンドウの設定

${id}	1 から 5 までの、参考訳文の番号
${sourceText}	参考訳文の原文
${targetText}	参考訳文の訳文
${diff}	原文と参考訳文との差異を表示します。ヒント：更新された文書の翻訳時に使うとよいでしょう。
${score}	トークナイザーの一致率
${noStemScore}	数字とタグを除いた一致率
${adjustedScore}	補正した一致率
${fileNameOnly}	TMX ファイル名
${filePath}	TMX ファイルのフルパス
${fileShortPath}	tm フォルダーを起点にした TMX ファイルの相対パス
${creationID}	参考訳文の翻訳者
${creationDate}	参考訳文の更新日
${fuzzyFlag}	参考訳文がファジーであることを示す (今のところ #fuzzy マークのついた PO ファイルの翻訳でのみ有効)

2.3. 用語集ウィンドウ

用語集ウィンドウには、用語集ファイルにあらかじめ蓄積した、表現や専門用語の対訳を表示できます。現在の分節中に見つかった用語を表示します。下図の原文は「Context menu command」ですが、そこに見つかった用語を用語集から表示しています。

図4.4 用語集ウィンドウ

用語ヒントを有効化している場合（［設定］→［用語ヒント］）は、原文分節中の下線が引かれた用語の上で右クリックすると、ポップアップメニューにその対訳が表示されます。対訳を選択すると、現在の翻訳入力行のカーソル位置に挿入されます。また、用語集ウィンドウから任意のテキストを選択して、選択領域を右クリックしても、同じように挿入できます。

2.4. 辞書ウィンドウ

辞書は、印刷物の辞書の電子版です。Merriam Webster、Duden、Larousse などの卓上にある辞書と同じ用途です。詳しくは「辞書」の章を参照してください。

2.5. 複数訳文ウィンドウ

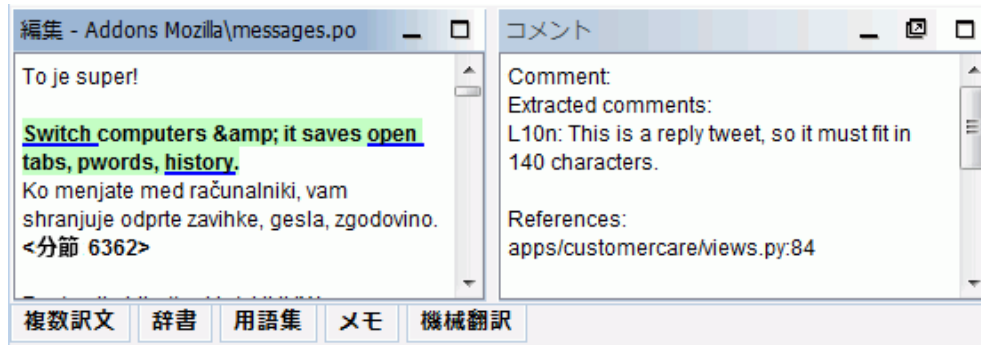
同じ内容の原文分節が複数存在し、それぞれの文脈に応じて異なる対訳が必要になることがあります。もし現在の翻訳がしっくりこない場合は、［編集］→［新しい訳文を登録］コマンドを選んでください。その後に入力した訳文は、「新しい訳文」として扱われます。同じ内容の他の原文に対する既存の訳文には影響しません。また、「新しい訳文」のうちの（もっとも普遍的な）1つを、既定値に設定することもできます。この場合は、［編集］→［既定値訳文として使用］を選んでください。

2.6. メモウィンドウ

翻訳者は現在の分節にメモを添付することができます。後で現在の分節に戻って翻訳しなおしたい場合や、複数訳文の他の値を確認してまわる場合、同僚に意見を聞きたい場合などに使います。メモを見るには、［移動］→［次のメモ］か［移動］→［前のメモ］を使います。

2.7. コメントウィンドウ

ファイルの中には、翻訳作業向けに特化した形式があります。たとえば PO ファイルは、翻訳者向けのコメント情報を格納できます。これによって翻訳者は、翻訳する分節に関する文脈を知ることができます。下記の例では、翻訳者向けに書かれた、訳文の文字数制限に関する注意事項を、コメントウィンドウに表示しています。

図4.5 コメントウィンドウ

2.8. 機械翻訳ウィンドウ

機械翻訳ウィンドウには、現在の分節に対応した機械翻訳による翻訳内容を自動的に表示できます。［Ctrl+M］を押すと、現在の分節のうち選択された部分を、機械翻訳に置換します。詳しくは「機械翻訳」の章を参照してください。

2.9. メインメニュー

メインメニューからは、OmegaT の全機能を使用できます。すべてのメニュー項目の詳細については、「メニューとキーボードショートカット」ページのメインメニューの各項目を参照してください。よく使われる機能についてはキーボードショートカットを使用できます。慣れてしまえば、翻訳作業中にメニューをわざわざたどる必要はなくなります。詳しくは「メニューとキーボードショートカット」を参照してください。

2.10. ステータスバー

メインウィンドウの底部にあるステータスバーには、実行中の処理についてのメッセージが表示されます。現在行われている処理の状態や結果などのフィードバックを知ることができます。現在の分節に対し、参考訳文の一致数と、用語集の用語との一致数も表示されます。

3. その他のウィンドウ

3.1. 翻訳対象ファイル一覧

翻訳対象ファイル一覧ウィンドウには、プロジェクトに含まれるファイル一式と、プロジェクトに関する情報が表示されます。OmegaT がプロジェクトを読み込むと、翻訳対象ファイル一覧が自動的に表示されます。［Ctrl+L］で開き、［Esc］で閉じます。翻訳対象ファイル一覧ウィンドウは、次の情報を表示します：

- プロジェクト中の翻訳可能なファイル数。対象は、原文ファイルフォルダーに存在し、OmegaT が扱える形式のファイルです。この数は、「翻訳対象ファイル一覧」のタイトルに続く括弧内に表示されます。
- プロジェクト中の翻訳可能なファイルの全リスト。クリックすると、翻訳作業のために開くことができます。注意：ヘッダーをクリックすると、たとえばファイル名（左端の列）のアルファベット順で並べ替えたりできます。また［...へ移動］ボタンでファイルの順番を任意に変更することもできます。
- 現在、編集ウィンドウで開かれているファイルは、青い背景色でハイライト表示されます。［Enter］キーを押すと、編集ウィンドウに遷移して、選択したファイルの先頭に移動します。
- 各ファイル項目には、ファイル名、ファイルフィルターの形式、エンコーディング、分節数が表示されます。
- 表の最下部には、「すべての分節数」と「繰り返しを除いた分節数」、「繰り返しを除いた翻訳済み分節数」が表示されます。

繰り返しを除いた分節数とは、すべての分節から重複するものを除いて数えた合計です。ここでいう「重複」は、大文字と小文字を区別します。「Run」と「run」は別物として数えます。

「すべての分節数」と「繰り返しを除いた分節数」の差から、すべての文中に繰り返しの表現がどれくらいあるかを大まかに見積もることができます。ただし、その数から、繰り返しの内容にどのていど強い相関があるかを推し量ることはできません。とても長い文章が何度も繰り返し登場する場合もありますし（おめでとうございます）、短いキーワードが並んだ表が含まれているだけの場合もあります（気を落とさずに）。omegat フォルダーにある project_stats.txt には、各ファイルに含まれる分節について の詳細な情報が記載されています。

分節化規則を修正すると、「すべての分節数」と「繰り返しを除いた分節数」が変わる可能性があります。しかし、いったんプロジェクトの翻訳を開始したら、規則の変更はなるべく行わないようにしてください。詳しくは「分節化規則」の章を参照してください。

プロジェクトへのファイル追加：［原文ファイル追加...］ボタンをクリックして、原文ファイルを追加できます。選択したファイルは source フォルダーへ追加され、プロジェクトが再読み込みされます。［MediaWiki から原文ファイルを追加...］ボタンを押して URL を指定すると、MediaWiki 形式で記述されたウェブページを原文ファイルとして追加することができます。

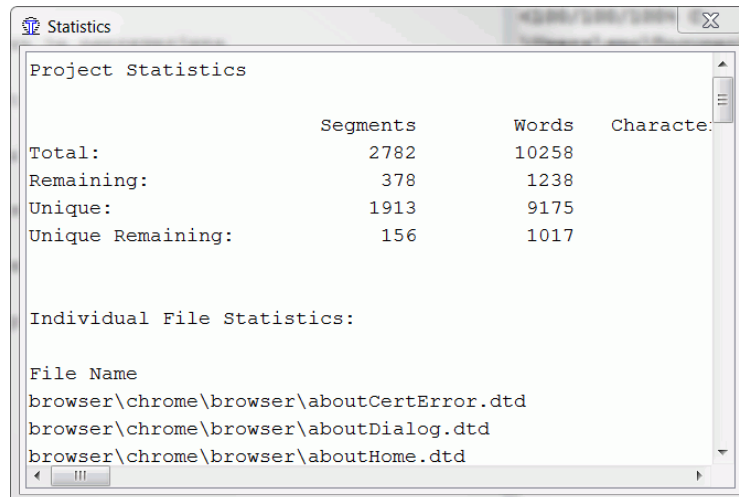
3.2. 検索ウィンドウ

プロジェクト内のある分節を見つけるために、検索ウィンドウを活用できます。同時に複数の検索ウィンドウを開くこともできます。新しい検索ウィンドウは、メインウィンドウ上で［Ctrl+F］を押すと開きます。検索ウィンドウは、検索文字列またはキーワードを入力するテキスト領域と、検索の設定を行うラジオボタンやチェックボックス、そしてその下の検索結果を表示する領域からなります。検索ウィンドウの詳細については「検索」の章を参照してください。

3.3. タグ検証

タグ検証ウィンドウには、訳文中のタグの間違いや不整合の可能性がある分節の一覧が表示されます。［Ctrl+T］でウィンドウを表示できます。ウィンドウは 3 列の表からなり、リンク付きの分節番号、原文、訳文の内容がそれぞれ表示されます：

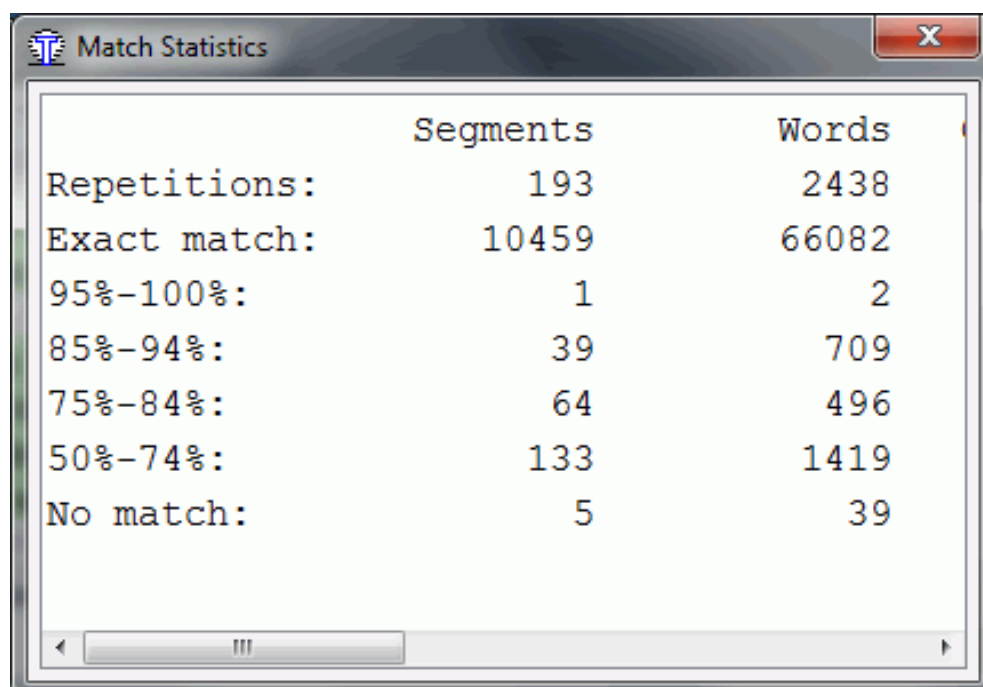
図4.6 タグ検証ウィンドウ

1243	(At this point, all text outside the target field is protected and cannot be modified.) You must type your translation between the tags **<c0>**<segment 0001>**</c0>** and **<c1>**<end segment>**</c1>**, overwriting the source text.	(Nun wird der ganze Text außerhalb des Zielfeldes geschützt und kann nicht überschrieben werden.) Die Übersetzung wird zwischen den Tags **<c1>**<segment 0001>**</c0>** und **<c0>**<end segment>**</c1>** eingetippt und dabei wird der Quelltext überschrieben.
1327	To display the User Manual, press "F1" or use the menu: **<s0>**Help → User Manual**</s0>**.	Um das Benutzerbuch zu zeigen, drücken Sie "F1" oder benutzen Sie Menü: s0>Hilfe {\lang1024 →} Benutzerhandbuch**</s0>**. &U
1383	(At this point, all text outside the target field is protected and cannot be modified.) You must type your translation between the tags **<c0>**<segment 0001>**</c0>** and **<c1>**<end segment>**</c1>**, overwriting the source text.	(Nun wird der ganze Text außerhalb des Zielfeldes geschützt und kann nicht überschrieben werden.) Die Übersetzung wird zwischen den Tags **<c1>**<segment 0001>**</c0>** und **<c0>**<end segment>**</c1>** eingetippt und dabei wird der Quelltext überschrieben.

原文と訳文で比較しやすいよう、タグは青い太字で強調表示されます。分節番号をクリックすると、編集ウィンドウにその分節を表示できます。必要に応じて訳文を修正し、［Ctrl+T］を押すと再度タグ検証ウィンドウに戻り、その他のエラーを表示します。上図の例のうち、１番目と３番目はタグの組み合わせが間違っている例で、２番目は開始タグの < 文字が抜けている例です。

タグエラーは、タグ操作によって、原文の分節にあるタグの登場順序や数が、訳文と一致しない場合におこります。タグ操作には、必要なもの、行っても問題ないものもありますが、中には翻訳後の文書を作成した際に問題となるものもあります。タグがあるということは、多くの場合、ある種の書式整形がそこで行われているということを表します。原文テキストの整形を単純にしておくと、タグは減ります。

3.4. 翻訳状況

翻訳状況ウィンドウは、［ツール］→［翻訳状況］から開きます。現在のプロジェクトの翻訳状況について、全体の概要とファイル単位の詳細な統計情報の両方を表示します。翻訳状況は omegat フォルダーに project_stats.txt ファイルとしても出力されます。タブ区切り形式になっており、任意の表計算プログラムで開くことができます。またウィンドウの内容は ［Ctrl+A］、［Ctrl+C］、［Ctrl+V］でコピー/貼り付けできます。

図4.7 プロジェクトの翻訳状況

3.5. 参考訳文の一致率

参考訳文の一致率は、［ツール］→［参考訳文の一致率］から開きます。値の計算には CPU を多く使用し、時間がかかることがあります。そのため、処理中にはプログレスバーが表示されます。一致率は、事実上の業界標準にしたがい、次のグループに分類されます：繰り返し、完全に一致、95% - 100%、85% - 94%、75% - 84%、50% - 74%、一致しない。この情報は、分節数、単語数、文字数（空白を除く）、文字数（空白を含む）それぞれについて、計算されます。OmegaT と他の翻訳支援ツールでは、一致率に相違が見られる場合がありますので、注意してください。

図4.8 参考訳文の一致率

これらの合計値は、そのプロジェクトに必要な作業を示す、かなり（少なくとも可能な限り）良い指標になります。これを、翻訳にかかる経費や料金を見積もるための基礎資料として使うことができるでしょう。

最終列の「文字数（空白を含む）」の値には、分節と分節のあいだの空白文字は含まれていません。「繰り返し」は、同一内容の分節が何度も使用されている場合を表しています。最初の分節とその内容は「一致しない」に分類されます。そして残りは、最初の分節の「繰り返し」として分類されます。プロジェクトの翻訳メモリの中に、同一の複数分節に対応する翻訳が存在する場合、これらの分節は、他の翻訳済みの繰り返しの無い分節と一緒に、「完全に一致」に分類されます。繰り返しの無い分節数が必要であれば、翻訳済みか否かにかかわらず、翻訳状況ウィンドウの方で確認できます。

残りのグループ（50 - 100%）は、参考訳文を伴う未翻訳の分節から構成されます。「繰り返し」と「完全に一致」に対する参考訳文は、omegat フォルダーの内部翻訳メモリのみならず、tm フォルダーの外部補助メモリも参照します。内部翻訳メモリである project_save.tmx との唯一の違いは、外部補助メモリの方は「完全に一致」ではなく「100%」に分類されるという点です。外部補助メモリを一致率に含めたくない場合は、tm フォルダーを空にしておくか、またはプロジェクトの設定で翻訳メモリフォルダーを（一時的に）別の場所に変更してください。

参考訳文の一致率はタブ区切り形式になっており、［Ctrl+A］、［Ctrl+C］、［Ctrl+V］でコピー/貼り付けして、任意の表計算プログラムや原価計算プログラムで利用することができます。いったん計算が終了すると、そのデータは omegat/project_stats_match.txt としてファイルに保存されます。ファイルの中には、タイムスタンプも出力されています。翻訳状況と違い、参考訳文の一致率はリアルタイムで計算されないので、すぐに古くなってしまいます。利用する際は、タイムスタンプをよく確認してください。

3.6. ヘルプウィンドウ

この取扱説明書を表示するには、［F1］を押すか、メニューから［ヘルプ］→［取扱説明書...］を選びます。ウィンドウには［戻る］と［目次］の２つのボタンがあります。この取扱説明書は、章から章へリンクした HTML 形式の文書です。ウェブブラウザー上での操作と同様に、各リンクをクリックすると対応したページを表示できます。

取扱説明書は OmegaT のインストールフォルダー内の docs フォルダーにあります。たとえば英語版を見たければ docs/en/index.html ファイルをウェブブラウザーで開けば閲覧できます。こうすることで、OmegaT 組み込みのヘルプブラウザーからは開けない外部サイトへのリンクも、たどって開くことができます。

第5章 メニューとキーボードショートカット

1. メインメニュー

OmegaT のすべての機能は、編集ウィンドウ上部にあるメニューから実行できます。多くの機能にキーボードショートカットが割り当てられており、それを用いて実行することもできます。ショートカットは [Ctrl] キーと文字キーを同時に押します。他のキーを含めて使用するものもあります。判別しやすいよう、文字キーは大文字で表記します。 [Ctrl] キーは Windows で使用します。UNIX や UNIX 系のオペレーティングシステムでは [Ctrl] または [Control] キーが該当します。Mac ユーザーは [cmd] +キーと読み替えてください。Apple キーボードでは、 [cmd] キーには「command」またはリンゴ マークが記されています。

必要であれば、既存のショートカットをカスタマイズしたり、新しいショートカットを追加することもできます。詳しくは付録 - ショートカットのカスタマイズを参照してください。

表5.1 メインメニュー

プロジェクト	編集	移動	表示	ツール	設定	ヘルプ

1.1. プロジェクト

表5.2 プロジェクトメニュー

新規作成...	Ctrl+Shift+N	新しいプロジェクトを作成し、開きます。ここで表示されるウィンドウは [プロジェクト設定] ダイアログ（詳細は右リンク先を参照）と同じです。6章プロジェクト設定
チームプロジェクトをダウンロード...		リモートの OmegaT プロジェクトのローカルコピーを作成します。
開く...	Ctrl+O	すでに作成したプロジェクトを開きます。
最近使用したプロジェクト		最近使用したプロジェクト（デフォルト設定では 5 つまで）にアクセスできます。選択肢のいずれかをクリックすると、現在のプロジェクトを保存して閉じ、選択したプロジェクトを開きます。
原文ファイルを追加...		選択したファイルをプロジェクトの source フォルダーにコピーし、その新しいファイルを読み込むために、プロジェクトを再読み込みします。
MediaWiki から原文ファイルを追加...		入力した URL をもとに MediaWiki 形式のページを開き、編集可能な部分を原文として追加します。
再読み込み	F5	外部からの原文ファイルの変更や、過去資産の翻訳メモリ、用語集、プロジェクト設定の変更を認識させるために、プロジェクト全体を再読み込みします。

閉じる	Ctrl+Shift+W	それまでの翻訳内容を保存し、プロジェクトを閉じます。
保存	Ctrl+S	内部翻訳メモリをハードディスクに保存します。OmegaT　はデフォルトで 10 分ごとに翻訳を自動保存します。プロジェクトを閉じる時や、OmegaT　の終了時も同様です。
訳文ファイルを生成	Ctrl+D	それまで翻訳した内容をもとに訳文ファイルを生成します。訳文ファイルは、target フォルダーに保存されます。
現在の訳文ファイルのみ生成	Ctrl+D	翻訳中の現在の文書に対応する訳文ファイルを生成します。
プロジェクト設定...	Ctrl+E	プロジェクトの言語や使用フォルダーを変更するためプロジェクト設定ダイアログを表示します。
翻訳対象ファイル一覧...	Ctrl+L	翻訳対象ファイル一覧ウィンドウを開くか、あるいは閉じます（どの動作をするかは、その時点でウィンドウが開いているかどうかによります）。
終了	Ctrl+Q	プロジェクトを保存し　OmegaT を終了します。まだ一度も手動で保存していなかった場合は、本当に閉じてもよいかどうか確認メッセージが表示されます。

1.2. 編集

多くのプログラムで使える機能（コピー、切り取り、貼り付け）は、このメニューに表示されていませんが、システムのショートカットとして OmegaT でも使用できます。たとえば：

表5.3 コピー / 切り取り / 貼り付けのショートカット

コピー	Ctrl+C	選択したテキストをクリップボードにコピーします。
切り取り	Ctrl+X	選択したテキストを削除し、クリップボードにコピーします。
貼り付け	Ctrl+V	現在のカーソル位置に、クリップボードにあるテキストを貼り付けます。

編集メニューに存在するのは以下の項目です：

表5.4 編集メニュー

取り消す	Ctrl+Z	直前の編集が行われる前の状態に戻します。分節の編集が確定された後の場合は、機能しません。
やり直す	Ctrl+Y	直前の編集が取り消される前の状態に戻します。分節の編集が確定された後の場合は、機能しません。

参考訳文に置換	Ctrl+R	現在の訳文分節の全体を、選択されている参考訳文で置き換えます（デフォルトでは最初の参考訳文が選択されています）。
参考訳文を挿入	Ctrl+I	現在のカーソル位置に、選択されている参考訳文を挿入します。もし訳文の一部を選択した状態であれば、その部分を上書きします。
機械翻訳に置換	Ctrl+M	訳文分節を選択した機械翻訳サービスによる翻訳内容に置換します。機械翻訳サービスが有効になっていない場合は、何も行いません（メニューの［設定］→［機械翻訳］を確認してください）。
原文に置換	Ctrl+Shift+R	現在の訳文分節を、原文の内容に置換します。
原文を挿入	Ctrl+Shift+I	現在の訳文分節のカーソル位置に、原文の内容を挿入します。
未使用の原文のタグを挿入	Ctrl+Shift+T	現在の訳文分節のカーソル位置に、訳文に未使用の原文のタグを挿入します。
次の未使用のタグを挿入	Ctrl+T	現在の訳文分節のカーソル位置に、訳文に未使用の原文のタグを 1 つだけ挿入します。
選択領域を出力	Ctrl+Shift+C	選択した領域を別のテキストファイルに出力します。選択領域がない場合、現在開いている原文がこのファイルに書き込まれます。OmegaT　を閉じるとき、このファイルは空の状態ではなく、通常のクリップボードと同じように機能します。出力された内容は、個人設定フォルダー（「8章OmegaT ファイルとフォルダー」を参照してください）の selection.txt というファイルにコピーされます。
用語の追加	Ctrl+Shift+G	デフォルトの用語集ファイルに項目を追加します。
プロジェクトを検索...	Ctrl+F	［検索］ウィンドウを新しく開きます。
検索と置換...	Ctrl+K	［検索と置換］ウィンドウを新しく開きます。
大/小文字設定	Shift+F3（右を参照）	訳文分節中の選択したテキストに対し、ケースを変更します（すべて小文字→すべて大文字→Title　case：単語先頭のみ大文字）。［Shift+F3］で、上記 3 通りを順に切り替えて実行します。テキストを何も選択していない場合、OmegaT　はカーソル右にある単語をその場で選択して実行します。

参考訳文を選択	Ctrl+#N	参考訳文ウィンドウに表示された N 番目の参考訳文を選択します（#N は 1 から 5 の数字）。選択された参考訳文は、現在の分節に対して置換または挿入できます。「参考訳文ウィンドウ」に、色表示を含めた詳細の説明があります。
既定値訳文として使用		現在の原文分節に対して複数の対訳が存在し、かつ現在の訳文分節が既定値ではない場合に、現在の訳文分節を既定値に設定します。現在の原文分節に対して 1 つの対訳しか存在しない場合は、この項目はグレーアウトして選択できません。
新しい訳文を登録		同じ内容の原文分節が複数存在し、それぞれの文脈に応じて異なる対訳が必要になることがあります。このメニュー項目を選ぶと、その訳文は現在の分節固有のものになります。同じ内容の他の原文分節には影響を与えません。
訳文を削除		訳文を削除し、分節を未翻訳の状態にします。
訳文を<空訳文>にする		訳文を空の状態と定義します。訳文ファイルを生成した場合、該当する分節には何も出力されません。編集ウィンドウ上では <空訳文> と表示されます。
原文と同一訳文を登録	Ctrl+Shift+S	原文と同じ訳文を登録できます。［設定］→［翻訳入力行...］の［原文と同じ訳文を許可］にチェックが入っていない場合でも、有効です。

1.3. 移動

表5.5 移動メニュー

次の未翻訳分節	Ctrl+U	プロジェクトの翻訳メモリにまだ存在していない（未翻訳の）次の分節に移動します。
次の翻訳済み分節	Ctrl+Shift+U	未翻訳分節を飛ばして、次の翻訳済み分節に移動します。
次の分節	Ctrl+N または Enter	次の分節に移動します。現在の分節がファイル末尾である場合は、次のファイルの最初の分節へ移動します。
前の分節	Ctrl+P or Ctrl+Enter	一つ前の分節に移動します。現在の分節がファイル先頭である場合は、前のファイルの末尾の分節へ移動します。
分節番号を指定...	Ctrl+J	指定した番号の分節を開きます。

次のメモ		メモが添付された次の分節を開きます。
前のメモ		メモが添付された前の分節を開きます。
選択中の参考訳文の元分節	Ctrl+Shift+M	参考訳文ウィンドウ内で選択されている参考訳文に対応する分節へ移動します。
分節の選択履歴を進む	Ctrl+Shift+N	OmegaT　は分節を選択した履歴を記憶しています。このコマンドを実行すると、［ 分節の選択履歴を戻る ］コマンドによって移動した履歴をたどり、戻る前に選択していた分節へ移動できます。
分節の選択履歴を戻る	Ctrl+Shift+P	１つ前に選択していた分節へ戻ることができます。後で現在の分節に復帰するには、上にある［ 分節の選択履歴を進む ］コマンドを使います。

1.4. 表示

表5.6 表示メニュー

翻訳済み分節を色づけ	チェックがある場合、翻訳済みの分節が黄色でハイライト表示されます。
未翻訳分節を色づけ	チェックがある場合、未翻訳の分節が紫色でハイライト表示されます。
原文を表示	チェックがある場合、各分節に対応した原文が緑色にハイライト表示されます。チェックがない場合、原文の分節は表示されません。
繰り返しのある分節を色づけ	チェックがある場合、繰り返しのある分節が淡い灰色の文字色で表示されます。
メモ付き分節を色付け	チェックがある場合、メモ付きの分節が水色でハイライト表示されます。この色付けは、［ 翻訳済み分節を色づけ ］や［ 未翻訳分節を色づけ ］よりも優先度が上です。
ノーブレークスペースを色付け	チェックがある場合、ノーブレークスペース（ 自動改行しないスペース文字 ）が灰色の背景色で表示されます。
空白文字を色付け	チェックがある場合、スペース文字が小さな点で表示されます。
双方向アルゴリズム制御文字を色付け	チェックがある場合、双方向アルゴリズム制御文字 [http://www.w3.org/International/questions/qa-bidi-controls]を表示します。
auto フォルダー由来の分節を色付け	チェックがある場合、（ たとえば /tm/auto フォルダー内の TMX ファイルから ）自動挿入された全分節の背景を色付けします。色付けは、［ 設定 ］→［ 翻訳入力行... ］の［ auto　フォルダー由来の翻訳かどうかを記録する ］にチェックが入っている場合に限り、有効です。auto フォルダーから挿入された通常の翻訳は、オレンジ色で表示されます。それ以外の翻訳は、TMX の種類に応じた異なる色で表示されます。技術的な詳細は、機能改

	善リクエスト [http://sourceforge.net/p/omegat/feature-requests/963/]を参照してください。
更新情報を表示	更新情報の表示に関するオプション設定です。［現在の分節で表示］を選ぶと、現在の分節を最後に更新した作業者と更新日時の情報を表示します。［すべての分節で表示」を選ぶと、すべての分節について同様の情報を表示します。［表示しない」を選ぶと、何も表示しません。

注：色は配色の［設定］→［配色のカスタマイズ...］ダイアログから変更できます。

1.5. ツール

表5.7 ツールメニュー

タグ検証	Ctrl+Shift+V：整形されたファイルの中で、タグが欠けていたり、位置が変わったりしている箇所がないかを調べます。タグの間違いや不整合の可能性がある分節の一覧が表示されます。詳しくは「タグ検証」および12章整形されたテキストを扱うを参照してください。
タグ検証（現在の文書のみ）	上記と同じ処理を、翻訳中の現在文書についてのみ実行します。
翻訳状況	新しいウィンドウに、プロジェクトの翻訳状況を表示します。これには、プロジェクト全体の翻訳状況と、ファイルごとの翻訳状況が含まれます。
参考訳文の一致率	プロジェクトにおける参考訳文の一致率を表示します。これは繰り返し、完全に一致、部分的に一致、一致しないといった基準で分類されており、それぞれについて分節数、単語数、文字数を集計しています。
参考訳文の一致率（ファイル別）	プロジェクトにおける参考訳文の一致率をファイル別に表示します。これは繰り返し、完全に一致、部分的に一致、一致しないといった基準で分類されており、それぞれについて分節数、単語数、文字数を集計しています。
スクリプト...	スクリプト管理用のウィンドウを表示します。このウィンドウ上で、スクリプトフォルダーの指定、スクリプトの編集や実行、ショートカットへの関連付けを行えます。詳しくはスクリプト ウィンドウを参照してください。

1.6. 設定

表5.8 設定メニュー

TAB キーで次の分節へ	編集する分節を次へ進めるキーを、デフォルトの［Enter］キーの代わりに［Tab］キーとします。この設定は、中国語、日本語、または朝鮮語のような入力システムの場合に便利です。
終了時に常に確認	OmegaT を終了する前に、常に確認ダイアログを表示します。
機械翻訳	機械翻訳ツールを有効または無効にします。有効な場合、［Ctrl+M］を押すと、現在の分節に機械翻訳の結果を挿入します。

用語集	TBX 用語集の説明文を表示オプションは、用語（用語集フォルダー内の TBX ファイル由来）を含む文脈情報の表示を切り替えます。 これ以外の 3 つのオプションについては、TaaS（Terminology as a Service）に関する専用ページで説明しています。
用語ヒント	用語ヒント機能と、そのオプションの完全一致を有効または無効にします。用語ヒントが有効な状態では、原文分節に含まれる用語に下線が引かれ、その上で右クリックするとポップアップメニューに用語の対訳が表示されます。対訳をクリックすると、訳文分節のカーソル位置に対訳が挿入されます。［用語ヒント］→［完全一致］にチェックを入れると、語句が完全に一致する用語のみ表示されます。チェックを入れないと、部分一致する用語も表示されます。
入力補完	［用語集...］をクリックすると、入力補完の用語集ビューの設定を変更できます。 ［定型文...］をクリックすると、定型文の設定変更や、定型文の追加と削除ができます。 ［文字一覧...］をクリックすると、文字一覧の設定を変更できます。
字体...	表示用の字体を設定するダイアログを表示します。古いコンピュータを使用しており、ウィンドウのサイズ変更が非常に遅いと感じるユーザーは、字体の変更を試してください。詳しくは「その他」の「字体の設定」を参照してください。
配色のカスタマイズ...	ユーザーインターフェースの各部分について、任意の色に変更できます。 定義済みのテーマを簡単に切り替えられるスクリプトも、付属しています。Switch Colour Theme スクリプトを使うと、既定で「ダーク」テーマを設定できます。
ファイルフィルター...	ファイルの取り扱いを設定する［ファイルフィルター］ダイアログを表示します。
分節化規則...	テキストの分節化を設定する［分節化規則の設定］ダイアログを表示します。
綴り確認...	綴り確認機能を組み込み、設定するための［綴り確認機能の設定］ダイアログを表示します。
翻訳入力行...	［翻訳入力行の設定］ダイアログを表示します。
タグ検証...	プログラマーの方へ：タグ検証機能はオプションで、プログラミング用の変数（書式指定子）も検証するよう設定することができます。
チーム...	翻訳した分節の更新情報に記録する更新者名を入力します。
外部 TMX ファイル...	外部の（OmegaT が生成したものではない）TMX ファイルのタグをどのように扱うか、設定します。 また参考訳文の並び替え基準を変更することもできます（結果表示のみの変更です。参考訳文の検索処理には影響しません）。参考訳文表示テンプレート欄を使うと、テンプレート変数を組み合わ

	せることによって、参考訳文をどのように表示するかカスタマイズすることができます。
表示...	テキストおよび更新情報の表示に関するオプションを設定できます。
保存と訳文ファイル生成	プロジェクトを自動的に保存する間隔を、分と秒で指定します。設定可能な最小間隔は、10　秒です。 また同じダイアログボックス上で、後処理用外部コマンド（訳文ファイルを生成した後に実行されます）も設定できます。
プロキシ認証...	プロキシサーバーを使う場合は、ここでユーザー名とパスワードを設定します。
ウインドウを初期位置に復元	OmegaT　で使用する各ウィンドウを、初期位置に戻します。切り離されたり、移動したり、隠れてしまったウィンドウを元の状態に戻したいときに、使用してください。また、OmegaT　をバージョンアップし、いくつかのウィンドウが表示されなくなったような場合にも、使用できます。
言語チェッカー	有効にすると、LanguageTool によって訳文が検査され、文法間違いの可能性のある箇所に青い下線が引かれるようになります。

1.7. ヘルプ

表5.9 ヘルプメニュー

取扱説明書...	［F1］キーを押すと、取扱説明書を独立したウィンドウで開きます。
OmegaT について...	著作権やクレジット、ライセンス情報を表示します。
最近の更新履歴...	各リリースごとに、新機能や改善、バグ修正などの情報を一覧で表示します。
ログ...	現在のログファイルを表示します。ダイアログのタイトルには、現在使用中のログファイル名（名称は同時に実行中の OmegaT のインスタンス数によって変わります）が表示されます。

2. キーボードショートカット

以下のショートカットは、メインウィンドウ上で使用できます。他のウィンドウが前面に表示されている場合、［Esc］を押してそのウィンドウを閉じるか、メインウィンドウを一度クリックして、前面に出してください。

［Ctrl］キーと文字キーを同時に押すと、ショートカットが動作します。他のキーを含めて使用するものもあります。判別しやすいよう、文字キーは大文字で表記します。

［Ctrl］キーは Windows で使用します。UNIX や UNIX 系のオペレーティングシステムでは［Ctrl］または［Control］キーが該当します。Mac ユーザーは［cmd］+キーと読み替えてください。Apple キーボードでは、［cmd］キーには「command」または リンゴ マークが記されています。

- プロジェクト
- 編集
- 移動

- ウィンドウの参照
- その他

2.1. プロジェクト

表5.10 プロジェクトのショートカット

開く	Ctrl+O	既存のプロジェクトを選択するウィンドウを表示します。
保存	Ctrl+S	現在の翻訳内容を、内部翻訳メモリ（プロジェクトの omegat フォルダーにある project_save.tmx）に保存します。
閉じる	Shift+Ctrl+W	現在のプロジェクトを閉じます。
訳文ファイルを生成	Ctrl+D	プロジェクトの target フォルダーに訳文ファイルを生成します。また翻訳メモリファイル（level1、level2 および omegat 形式の TMX ファイル）をプロジェクトのルートフォルダーに生成します。
プロジェクト設定	Ctrl+E	プロジェクト設定ダイアログを表示します。設定の変更も行えます。

2.2. 編集

表5.11 編集のショートカット

取り消す	Ctrl+Z	現在の訳文分節において、直前の編集が行われる前の状態に戻します。
やり直す	Ctrl+Y	現在の訳文分節において、直前の編集操作を再度行います。
参考訳文 #N を選択	Ctrl+#N	#N は 1 から 5 までの数字です。現在の分節に対して置換または挿入するために、参考訳文ウィンドウに表示された N 番目の候補を選択します。
参考訳文に置換	Ctrl+R	現在の訳文分節の内容を、選択されている参考訳文に置換します（デフォルトでは最初の参考訳文が選択されています）。
参考訳文を挿入	Ctrl+I	現在の訳文分節のカーソル位置に、選択されている参考訳文を挿入します（デフォルトでは最初の参考訳文が選択されています）。
原文に置換	Ctrl+Shift+R	現在の訳文分節を原文の内容に置換します。
原文を挿入	Ctrl+Shift+I	現在の訳文分節のカーソル位置に、原文の内容を挿入します。

原文のタグを挿入	Ctrl+Shift+T	現在の訳文分節のカーソル位置に、原文のタグを挿入します。
プロジェクトを検索...	Ctrl+F	検索ウィンドウを表示します。
機械翻訳に置換	Ctrl+M	現在の訳文分節を、機械翻訳エンジンによる機械翻訳に置換します。機械翻訳が有効になっていない場合は、何も行いません（メニューの［設定］→［機械翻訳］を確認してください）。
選択領域を出力	Shift+Ctrl+C	選択した領域を別のテキストファイルに出力します。
用語の追加	Shift+Ctrl+G	デフォルトの用語集ファイルに項目を追加します。

2.3. 移動

表5.12 移動のショートカット

次の未翻訳分節	Ctrl+U	編集入力行を、プロジェクトの翻訳メモリにまだ存在していない（未翻訳の）次の分節に移動させます。
次の分節	Ctrl+N、Enter または Return	翻訳入力行を、次の分節に移動させます。
前の分節	Ctrl+P	翻訳入力行を、１つ前の分節に移動させます。
分節番号を指定...	Ctrl+J	翻訳入力行を、指定した分節番号に移動させます。
分節の選択履歴を戻る	Ctrl+Shift+P	分節移動の履歴をたどり、１つ前に戻ります。
分節の選択履歴を進む	Ctrl+Shift+N	分節移動の履歴をたどり、１つ次に進みます。

2.4. その他

表5.13 その他のショートカット

翻訳対象ファイル一覧...	Ctrl+L	プロジェクト内の翻訳対象となるファイルを一覧表示します。
タグ検証	Ctrl+T	タグ検証ウィンドウを開きます。
選択領域を出力	Shift+Ctrl+C	選択領域（または何も選択されていなければ原文分節）を出力します。テキストはプレーンテキストファイルとして出力されます。
プロジェクトを検索...	Ctrl+F	新しい検索ウィンドウを開きます。
ヘルプ	F1	OmegaT　取扱説明書を別ウィンドウで表示します。

第6章 プロジェクト設定

1. 概要

プロジェクト → プロジェクト設定... (Ctrl+E) ダイアログは、プロジェクトの言語や使用フォルダーを変更する場合に使用します。

翻訳作業の途中でプロジェクトの設定を変更することも、可能ではありますが、そのような変更は、予期しない結果をもたらすことがあります。特に翻訳作業が始まってからの変更には、注意してください。OmegaT であるていど経験を積むまでは、一度翻訳を開始したら、何か致命的な誤りに気づいたのでない限り、設定を変えずにおくことがもっとも安全です。作業中のデータを守る対策については「データを失わないために」の項を参照してください。

2. 言語設定

原文言語と訳文言語は、リストから選択するか、または直接入力で指定します。もし翻訳途中で言語設定を変更すると、変更前後の言語間で一致が取れなくなり、翻訳メモリが使い物にならなくなることがあるので注意してください。

トークナイザーは、選択した言語に対応するものが表示されます。詳しくは、「付録 トークナイザー」を参照してください。

3. 設定

文単位の分節化を有効にする

分節化の設定は、原文ファイルの取り扱い方法にのみ関係します。原文の分節化は、文単位で行うのが一般的です。したがって通常は、このチェックボックスにチェックを入れておくのがよいでしょう。

めったにないことですが、段落単位での分節化のような、別の方法が必要とされることもあります。このフラグを変更しても、翻訳メモリにすでに記録された分節は変更されません。たとえば、翻訳の途中で分節化を文単位から段落単位に変更しても、そのプロジェクトの内部翻訳メモリは何も変わりません。OmegaT は段落単位でしか分節化できなかった旧形式の翻訳メモリをアップグレードできますが、その逆はできないのです。ただし OmegaT は、文単位の既存の翻訳を結合して、段落単位の参考訳文を作ることはできます。

分節化設定を変更した場合、翻訳済みの分節が分割されたり、結合されることがあります。これにより、たとえ分割や結合前の翻訳がまだ翻訳メモリに残っていたとしても、実質的に「未翻訳」の状態に戻ることになります。

分節化規則...

分節化規則は、ほとんどの場合、すべてのプロジェクトに共通で適用されます。しかし特定のプロジェクトに特化した分節化規則が必要になることもあるでしょう。その場合は、このボタンを押してダイアログを開き、［分節化規則をプロジェクト専用にする］チェックボックスにチェックを入れてから、分節化規則を必要な状態に変更してください。新しい規則集は、プロジェクトと共に保存されます。またプログラム共通の分節化規則集には干渉しません。プロジェクト専用の分節化規則を削除したい場合は、チェックボックスのチェックを外してください。分節化規則の詳細については「原文の分節化」の章を参照してください。

ヒント：プロジェクト専用の分節化規則集は project/omegat/segmentation.conf に保存されます。

ファイルフィルター...

前の項と同じように、プロジェクト専用のファイルフィルター設定を作成できます。この設定はプロジェクトと共に保存され、現在のプロジェクトにのみ適用されます。プロジェクト専用のファイルフィルター設定を作成するには、［ファイルフィルター...］ ボタンをクリックし、開いたウィンドウの［ファイルフィルター設定をプロジェクト専用にする］チェックボックスにチェックを入れてください。これにより、変更されたフィルター設定のコピーがプロジェクトと共に保存されるよ

	うにます。プロジェクト専用のファイルフィルター設定を削除するには、チェックボックスのチェックを外してください。［設定］→［ファイルフィルター...］メニューからユーザーレベルのファイルフィルター設定を変更しても、プロジェクト専用のファイルフィルター設定は変更されないことに注意してください。詳しくはファイルフィルターの章を参照してください。 ヒント：プロジェクト専用のファイルフィルター設定は project/omegat/filters.xml に保存されます。
翻訳自動反映	同じ内容の分節が何箇所か原文に存在する場合、［翻訳自動反映］チェックボックスにチェックを入れると、最初にその内容を一箇所翻訳した時点で、それが既定値訳文に設定され、他の訳文分節にも同じ内容が設定されます。原文が同じでも、文脈によっては他と同じ内容が適さない分節があるかもしれません。その場合はもちろん、後で［新しい訳文を登録］コマンドを使って手修正することができます。［翻訳自動反映］チェックボックスにチェックを入れないと、原文が同じ他の分節は未翻訳のまま残されます。通常の分節と同じように、その分節に移動して都度、翻訳することになります。
タグを削除	この設定にチェックを入れると、原文分節からすべての整形用のタグを削除します。これはインラインの整形を実際には使わないテキスト（たとえば OCR ソフトが出力した PDF ファイルや、変換精度の低い .odt ファイルや .docx ファイルなど）を扱う場合に役に立ちます。削除されるのはインラインのタグのみです。したがって通常は、訳文ファイルを開くのに特に支障は無いでしょう。非視覚的な整形（OmegaT の編集画面上には表示されません）は、訳文ファイルに引き継がれます。
後処理用外部コマンド	この欄には、訳文ファイルの生成後に実行したい外部コマンド（たとえば、ファイル名を変更するスクリプトなど）を記述できます。外部コマンドには「パイプ機能」を含めることができません。そのため、スクリプトファイルの単純な呼び出しが推奨されます。

4. ファイル場所設定

ここで、プロジェクトフォルダーの場所を変更することができます。たとえば、原文ファイルフォルダーや、訳文ファイルフォルダーなどです。フォルダー名を直接入力して、そのフォルダーがまだ存在しなければ、OmegaT がそのフォルダーを作成します。プロジェクトフォルダーの場所を変更しても、既存のファイルが古いフォルダーから新しいフォルダーに移動されたりはしませんので、注意してください。

［除外...］をクリックして、OmegaT に無視させたいファイルやフォルダーを登録できます。除外登録したファイルやフォルダーは、次のように扱われます。

- 編集ウィンドウに表示されません
- 翻訳状況の集計対象になりません
- 訳文ファイル生成時に、訳文ファイルフォルダーにコピーされません。

除外構成例集ダイアログで、構成例の追加と削除ができます。また既存の構成例を選んで F2 キーを押すと編集もできます。構成例には Ant 構文 [https://ant.apache.org/manual/dirtasks.html#patterns]にのっとったワイルドカード文字も使えます。

第7章 ファイルフィルター

OmegaT には高度にカスタマイズ可能なフィルター機能があり、さらにその多くの挙動設定を変更できます。ファイルフィルターは以下のような機能を持ったコードの集まりです：

- ある特定のファイル形式の文書を読み込みます。たとえばプレーンテキストファイルです。
- そのファイルから翻訳可能な部分を抽出します。
- 訳文ファイルの生成時に、原文分節を訳文に置き換えたり、規則に基づきファイル名を修正するなどの自動処理を行います。

OmegaT が対応しているファイル形式は、メニューの [設定] → [ファイルフィルター] から一覧できます。

ほとんどの場合は、デフォルトのファイルフィルター設定のままで問題ないでしょう。もしそうでなければ、メインメニューから [設定] → [ファイルフィルター] を選択してダイアログを開いてください。プロジェクト設定から [ファイルフィルター...] を選ぶことで、現在のプロジェクト専用のファイルフィルターを設定することもできます。この設定は、他のプロジェクトに影響を与えません。

プロジェクト専用ファイルフィルターを設定するには、まずメニューから [プロジェクト] → [プロジェクト設定...] を選びます。次に [ファイルフィルター...] ボタンをクリックして、 [ファイルフィルター設定をプロジェクト専用にする] チェックボックスにチェックを入れます。こうすると、フィルター設定のコピーがプロジェクト自体とセットで保持されるようになります。この状態でフィルターの設定を変更すると、プロジェクト専用フィルターだけが更新されます。ユーザーレベルのフィルターには影響しません。

警告：プロジェクトを開いているときにフィルター設定を変更すると、変更を適用するためにプロジェクトの再読み込みが必要になります。

1. ファイルフィルター ダイアログ

使用可能なファイルフィルターを一覧表示します。OmegaT の中で翻訳したくないファイル形式がある場合は、ファイル形式名の横にあるチェックボックスのチェックを外すと、その形式に対してフィルター機能を使わないようにすることができます。フィルターを無効にしているファイル形式のファイルは、プロジェクトの読み込み時に無視され、訳文生成時には未変更の状態で訳文フォルダーにコピーされます。再びフィルターを使用したくなったら、対象のチェックボックスにチェックを入れてください。 [初期値を復元] をクリックすると、それまでのフィルター設定をリセットし初期設定に戻します。フィルター対象とするファイルやエンコーディングを変更する場合は [編集...] をクリックします。

ダイアログ上で、次の設定の有効/無効を切り替えることができます：

- 先頭と末尾のタグを削除：先頭と末尾のタグを含むすべてのタグを表示したい場合は、この設定のチェックを外してください。警告：Microsoft Open XML 形式のファイル（docx や xlsx など）について、すべてのタグを表示した場合は、決して先頭のタグの前にテキストを入力しないでください（そのタグは技術的な理由で、常に分節の先頭に置かなければなりません）。
- 分節化しないプロジェクトから、先頭と末尾の空白文字を削除：OmegaT は既定で、先頭と末尾の空白文字を削除します。分節化しないプロジェクトの場合、このオプションのチェックを外すことによって、それらの空白文字を保持することができます。
- すべてのタグの空白を維持：原文ファイルにとって空白文字が（レイアウトの都合上）重要な意味を持ち、それを無視すべきでない場合は、このオプションのチェックを入れてください。
- 複数訳文の識別時にファイルのフルパス情報を無視する：OmegaT は既定で、原文ファイル名を複数訳文の識別基準のひとつに利用します。このオプションにチェックを入れると、ファイル名は識別に利用されません。複数訳文の識別には、ファイル名以外の文脈情報（前後の分節や、ファイル形式によってはある種の ID）だけが使用されます。

2. フィルター設定

以下のフィルター（テキストファイル、PO ファイル、XHTML ファイル、HTML ファイル･ XHTML ファイル、OpenDocument ファイル、Microsoft Open XML ファイル）では、いくつかの専用オプションを設定できます。設定を変更する場合は、フィルターを選択して［設定...］を押してください。可能な設定は以下の通りです：

テキストファイル

- 原文を改行で分節化するか、空行で分節化するか、分節化しないか：

  文単位の分節化が有効になっている場合は、ここで選択した設定によってさらに分節化されます。

PO ファイル

- 訳文ファイルの空分節を許可：

  この設定を有効にすると、未翻訳の PO 分節（たいていは段落全体）を空のままにして訳文ファイルを生成します。もう少し技術的に立ち入った話をすると、PO 訳文ファイルの msgstr 分節が空のままになる、ということです。PO ファイルの場合はこれが一般的な状態ですので、この設定はデフォルトで有効になっています。この設定を無効にすると、原文が訳文分節にコピーされます。

- PO ヘッダーを無視

  この設定にチェックを入れると、PO ヘッダーを翻訳対象外とみなして、原文のままにします。

- ヘッダーの「nplurals=整数; plural=式;」を自動で置換

  この設定にチェックを入れると、PO ファイルのヘッダーにある複数形指定を、訳文言語に合わせて自動変更します。

XHTML ファイル

- 以下の場合に HTML･ XHTML ファイルにエンコーディング宣言を追加または変更：訳文ファイルのエンコーディング文字セットを、原文ファイルのそれ（明示的に指定されたものであれ、暗黙に適用されたものであれ）とは違うものにしたいということは、よくあります。この設定を使うと、訳文ファイルにエンコーディング宣言を含めるかどうかを指定できます。たとえば、ファイルフィルターで訳文ファイルのエンコーディングが UTF-8 に指定されている場合、この設定によって、訳文ファイルに必ずその情報を含めるようにすることができます。

- 以下の属性を翻訳：選択した属性を編集ウィンドウで分節として表示できます。

- 以下のタグで分節化： HTML の
 タグがある場所を段落とみなして分節化することができます。

- 以下の正規表現に一致する文字列を翻訳しない：指定した正規表現に一致するテキストは、翻訳対象とせずスキップすることができます。スキップした箇所は、タグ検証では赤く表示されます。参考訳文ウィンドウでは、スキップした箇所が原文分節にあると、そのテキストは斜体で表示されます。

- <meta> タグ内の以下の属性と値は翻訳しない（カンマ区切りで複数指定）：ここで指定した <meta> タグは、翻訳されません。

- 以下の属性と値を持つタグの内容は翻訳しない（カンマ区切りで複数指定）：ここでキー = 値の形式で指定したタグは、翻訳されません。

  この設定は、ある属性の値に応じてタグを翻訳しないようにするのに、役に立つことがあります。たとえば、<div class="hide"> <span translate="no"> のようなタグは、キー = 値の形式で指定して翻訳対象外に設定できます。この例の場合、設定欄は次のようになります：class=hide, translate=no

Microsoft Office Open XML ファイル

どの要素を翻訳するか、選ぶことができます。選んだ要素は、翻訳時に個別の分節として表示されます。

- Word：非表示の命令とテキスト、コメント、脚注、文末脚注、ヘッダー、フッター
- Excel：コメント、シート名
- Power Point：スライドのコメント、スライド マスター、スライド レイアウト
- 共通：表、ダイアグラム、図、ワードアート
- その他の設定：
  - タグを結合：チェックが入っていれば、翻訳対象テキストを含まない連続するタグを 1 つのタグに結合します。
  - すべてのタグの空白を維持：チェックが入っていれば、空白文字（半角スペースや改行など）を保持します。たとえそれが技術的には意味が無い場合でも、残します。

HTML ファイル・XHTML ファイル

- 以下の場合に HTML・XHTML ファイルにエンコーディング宣言を追加または変更：常に（デフォルト）、(X)HTML ファイルにヘッダーがある場合のみ、(X)HTML ファイルにエンコーディング宣言がある場合のみ、しない
- 以下の属性を翻訳：選択した属性を編集ウィンドウで分節として表示できます。
- 以下のタグで分節化： HTML の
 タグがある場所を段落とみなして分節化することができます。
- 以下の正規表現に一致する文字列を翻訳しない：指定した正規表現に一致するテキストは、翻訳対象とせずスキップすることができます。
- <meta> タグ内の以下の属性と値は翻訳しない（カンマ区切りで複数指定）：ここで指定した <meta> タグは、翻訳されません。
- 以下の属性と値を持つタグの内容は翻訳しない（カンマ区切りで複数指定）：ここでキー＝値の形式で指定したタグは、翻訳されません。

テキストファイル

- 原文を改行で分節化するか、空行で分節化するか、分節化しないか：

  文単位の分節化が有効になっている場合は、ここで選択した設定によってさらに分節化されます。

OpenDocument Format (ODF) ファイル

- どの要素を翻訳するか、選ぶことができます：

  Index 項目、ブックマーク、ブックマーク参照、脚注、コメント、プレゼンテーションのノート、リンク（URL）、シート名

3. フィルターの編集ダイアログ

このダイアログでは、フィルターで処理する原文ファイル名の構成例の指定、訳文ファイル名のカスタマイズ、読み込む原文ファイルと生成する訳文ファイルのエンコーディングを設定できます。ファイルフィルター構成例を編集するには、それぞれの欄をダブルクリックして直接修正するか、編集ボタンをクリックします。新しいファイルフィルター構成例を追加するには追加ボタンをクリックします。構成例の追加と編集には、同じダイアログを使用します。このダイアログは、訳文ファイル名の構成例編集に特化したエディターを持っているため、出力するファイル名のカスタマイズに便利です。

3.1. 原文ファイル形式、ファイル名構成例

OmegaT は原文フォルダーにファイルを見つけると、まず拡張子から判別してファイルフィルターを選択しようとします。もっと具体的に言うと、各フィルターの原文ファイル名構成例から、ファイル名と一致するものを探索します。たとえば、構成例 *.xhtml は、拡張子が .xhtml のあらゆるファイルに一致します。適切なフィルターが見つかった時点で探索は終了し、そのフィルターが該当するファイルの処

理に割り当てられます。たとえばデフォルトでは、XHTML フィルターは拡張子 .xhtml のファイルを処理するのに使用されます。そのフィルターに扱わせたいファイルに応じて、ファイル名構成例を変更したり追加したりできます。原文ファイル名構成例には、検索機能と同様にワイルドカード文字を使用できます。ワイルドカード「*」は 0 個以上の任意の文字と一致します。ワイルドカード「?」は任意の 1 文字と一致します。他のすべての文字は、それらの文字自身をあらわします。たとえばテキストフィルターを readme ファイル（readme、read.me や readme.txt）にも対応させたい場合は、ファイル名構成例に read* を指定してください。（日本語訳注：実際にはワイルドカード文字と衝突しない範囲で、正規表現も使用できます。たとえばファイル名の末尾に数字のついた、拡張子 .foo のファイルを扱いたい場合は、*\d.foo のような指定が可能です。正規表現の詳細は、http://docs.oracle.com/javase/jp/6/api/java/util/regex/Pattern.html を参照してください）

3.2. 原文ファイルと訳文ファイルのエンコーディング

エンコーディングの指定が必須のファイル形式は限られています。エンコーディングを特に指定しなかった場合（<自動> の場合）は、拡張子に応じたエンコーディングが使用されます。たとえばデフォルトでは、.txt ファイルには OS のデフォルト エンコーディングが使用されます。原文ファイルのエンコーディングを、ファイル名構成例ごとに変更することもできます。まったく同様に、訳文ファイルを出力する際のエンコーディングも、任意に変更できます。デフォルトでは、訳文ファイルのエンコーディングは、原文ファイルと同じになります。原文ファイルと訳文ファイルのエンコーディングを指定する項目では、対応したすべてのエンコーディングをコンボボックスから選択できます。<自動> は、OmegaT に自動的に判断させます。自動の場合のエンコーディングは、次の仕組みで判定します：

- 原文ファイルにエンコーディング宣言があれば、それに従います（HTML ファイル、XML ベースのファイルの場合）
- いくつかのファイル形式（Java properties など）に対しては、その形式に必須のエンコーディングを設定します
- テキストファイルには、OS デフォルトのエンコーディング（たとえば日本語版の Windows OS では、シフト JIS 固定）を使用します

3.3. 訳文ファイル名

翻訳するファイル名を自動的に変更したい時があると思います。たとえば、ファイル名のあとに言語コードを追加したい場合などです。訳文ファイル名構成例は特別な文法で記述されています。これを編集したい場合は、［編集...］をクリックして構成例編集ダイアログを使用してください。編集内容をそのフィルターの初期値に戻したい時は、［初期値を復元］をクリックしてください。またファイルフィルターダイアログの中で、訳文ファイル名構成例を直接編集することもできます。構成例編集ダイアログでは、以下のような変数を使用できます：

- デフォルトは ${filename} です。-- 拡張子を伴った完全な原文ファイル名で、これを指定すると、訳文ファイル名は原文ファイル名と同じになります。
- ${nameOnly} -- 拡張子なしの原文ファイル名です。
- ${extension} -- 原文ファイルの拡張子のみです。
- ${targetLocale} -- 訳文のロケールコード（「xx_YY」形式）です。
- ${targetLanguage} -- 訳文の言語コードと国コードを合わせたもの（「XX-YY」形式）です。
- ${targetLanguageCode} -- 訳文の言語コードのみ（「XX」）です。
- ${targetCountryCode} -- 訳文の国コードのみ（「YY」）です。
- ${timestamp-????} -- 各種の書式による訳文ファイル生成時のシステム日時です。

  『SimpleDateFormat』の書式については、Oracle 技術文書 [http://docs.oracle.com/javase/jp/6/api/java/text/SimpleDateFormat.html]を参照してください。
- ${system-os-name} -- コンピューターの OS 名です。

- ${system-user-name} -- システムのユーザー名です。
- ${system-host-name} -- システムのホスト名です。
- ${file-source-encoding} -- 原文ファイルのエンコーディング名です。
- ${file-target-encoding} -- 訳文ファイルのエンコーディング名です。
- ${targetLocaleLCID} -- 訳文ファイルの Microsoft ロケール（「xx-yy」形式）です。

${nameOnly} と ${extension} に関しては、追加の変数を使用できます。ファイル名があいまいな場合、使用する構成要素を ${nameOnly-拡張子の個数} と ${extension-名前の個数} の形式で指定することができます。たとえば原文ファイル名が「Document.xx.docx」だったとしましょう。その場合、各種の変数とその結果は次のようになります：

- ${nameOnly-0} Document
- ${nameOnly-1} Document.xx
- ${nameOnly-2} Document.xx.docx
- ${extension-0} docx
- ${extension-1} xx.docx
- ${extension-2} Document.xx.docx

第8章 OmegaT ファイルとフォルダー

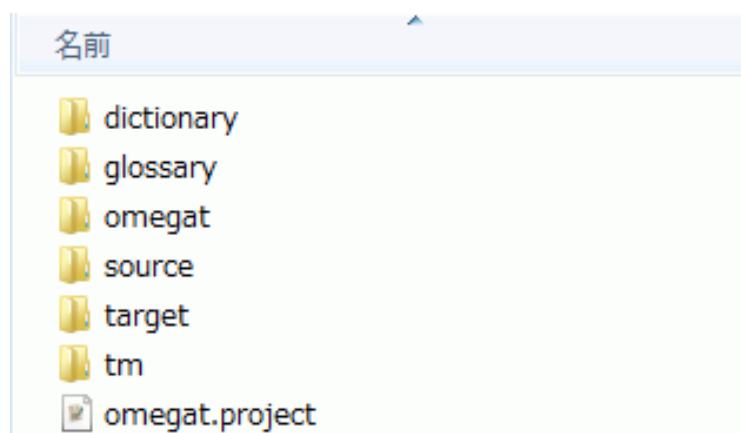
OmegaT は大きく分けて 3 種類のファイルを扱います。

- 翻訳プロジェクトファイル：翻訳プロジェクトを構成するファイル群です。これらが欠けると、プロジェクトの整合性が失われ、作業を完了できなくなることがあります。プロジェクトファイルはOmegaT の中でも最も重要です。翻訳中、常に使用することになるファイルです。
- 個人設定ファイル：OmegaT の設定がユーザーによって変更されたときに作成されるファイルです。これらのファイルが失われると、OmegaT　はいわゆる　「工場出荷時の（つまりデフォルトの）設定」に戻ることになります。この状態だと、翻訳途中で若干の問題が起きることもあります。
- プログラム　ファイル：ダウンロードしたパッケージに含まれるファイルです。ほとんどのファイルは、OmegaT を正常に動作させるために必要です。これらのファイルが何らかの理由で消失するか破損してしまった場合は、修復するために　OmegaT　を（必要であれば）再ダウンロードして、再インストールしてください。

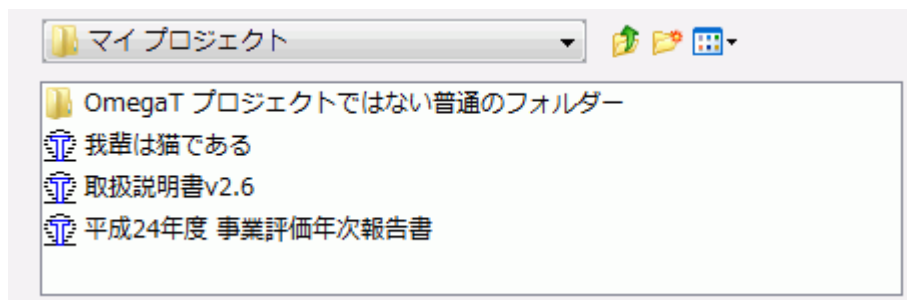
1. 翻訳プロジェクトファイル

OmegaT の翻訳プロジェクトは、複数のファイルとフォルダーから構成されています。

翻訳プロジェクトを作成すると、OmegaT は指定された名前のフォルダーを自動作成し、さらにその配下にいくつかのフォルダーを作成します：

図8.1 OmegaT プロジェクト

フォルダーの中には、プロジェクト作成時や翻訳作業の途中で、デフォルトから別の場所に変更できるものもあります。変更先のフォルダーは、各自のワークフローとプロジェクト管理習慣に応じて、既存の場所を指定したり、新規作成したりできます。プロジェクト作成後に場所を変更するときは、メニューから［プロジェクト］→［プロジェクト設定...］　を選ぶか、ショートカット［Ctrl+E］で設定ダイアログを開き、必要な変更を行ってください。

OmegaT 以外のファイルマネージャーで見ると、翻訳プロジェクトは他のフォルダーと特に区別無く表示されます。下の図は OmegaT での表示例ですが、マイ プロジェクトフォルダーに 3 つの OmegaT プロジェクトが含まれていることが分かります。

図8.2 OmegaT プロジェクトと通常のフォルダー

OmegaT アイコンをダブルクリックすると、プロジェクトが開きます。新規プロジェクトをデフォルトの設定で作成した場合、以下の構成で新しいフォルダー群が作成されます。

1.1. トップ フォルダー

トップ フォルダーには、必ず OmegaT.Project ファイルが存在します。このファイルには、プロジェクトの設定ウィンドウ（ [プロジェクト] → [プロジェクトの設定...] ）で設定された値が保存されています。翻訳作業を開始すると、このフォルダーにいくつかのファイル (プロジェクト名-omegat.tmx、プロジェクト名-level1.tmx、プロジェクト名-level2.tmx) が作成され、翻訳作業のあいだ絶えず更新されていきます。これらのファイルには、将来のプロジェクトで再利用できるように、まったく同じ翻訳メモリの同一の内容が、それぞれ異なる形式で記録されます。

1.2. dictionary フォルダー

このフォルダーは、最初は空の状態です。プロジェクトに辞書を追加する際、辞書ファイルの置き場所になります。詳しくは辞書の章を参照してください。

1.3. glossary フォルダー

このフォルダーは、最初は空の状態です。プロジェクトで使用する用語集ファイルの置き場所になります。詳しくは用語集の章を参照してください。

1.4. omegat フォルダー

omegat　フォルダーには最低でも 1 つ、場合によってはそれ以上のファイルが存在します。ここで最も重要なのは、project_save.tmx　でしょう。このファイルこそ、プロジェクトの作業用翻訳メモリになります。このファイルのバックアップ　ファイル (拡張子が　.bak) が、定期的に追加されていきます。保存されるタイミングは、翻訳作業の開始時、終了時、そして作業中など、随時です。これにより、データはどこへ消えた？という事態を未然に防ぐことができます。詳しくはその他の章に収録されている『データを失わないために』を参照してください。

他にも、翻訳作業中に必要に応じて以下のファイルが作成される場合があります。

stats.txt	現在のプロジェクトの翻訳状況が記録されます。メニューの [ツール] → [翻訳状況] から参照できます。
ignored_words.txt. learned_words.txt	綴り確認機能によって作成・使用されます。綴り確認機能に無視してほしい、あるいは学習させたい単語があれば、omegat フォルダーの対応する 2 ファイルに入力しておきます。
project_stats_match.txt	現在のプロジェクトの参考訳文の一致率が記録されます。メニューの [ツール] → [参考訳文の一致率] を選ぶと、作成されます。
segmentation.conf	このファイルが存在する場合は、 [プロジェクト] → [プロジェクトの設定...] で設定されたプロ

	ジェクト専用の分節化規則が含まれています。詳しくはプロジェクトの設定の章を参照してください。
filters.xml	このファイルが存在する場合は、［プロジェクト］→［プロジェクトの設定...］で設定されたプロジェクト専用のファイルフィルターが含まれています。詳しくはプロジェクトの設定の章を参照してください。

1.5. source フォルダー

source フォルダーには、翻訳したいファイルを置きます。ファイルは、後からでも追加できます。source フォルダー内部の階層構造をどうするかについては、完全にユーザーに任されています。翻訳したいファイルが（ウェブサイトのような）あるツリー構造の一部に入っている場合は、そのツリーの最上位のフォルダーを指定してください。そうすると OmegaT はツリー構造を保ったまま、全体を source フォルダーにコピーします。

1.6. target フォルダー

このフォルダーは、最初は空の状態です。メニュープロジェクト → 訳文ファイルを生成 (Ctrl+D)を選ぶと、この場所に訳文ファイルが生成されます。source フォルダーの中のファイルは、翻訳対象かどうかを問わずすべて生成されます。またフォルダーの階層構造もそのまま再現されます。生成されるファイルは、その時点の翻訳作業の進行状況を反映した内容になります。つまり、/omegat/project_save.tmx に保存された翻訳メモリの内容が反映されます。未翻訳分節は、原文のままです。

2. 個人設定ファイル

個人設定ファイルには、ユーザー固有の情報が格納されています。これはすべてのプロジェクト共通で使用されます。

logs/OmegaT.log	OmegaT 実行中の Java エラーメッセージが記録されます。もし OmegaT の動作がおかしいと思ったら、バグ報告の際に、このファイル自体または関連する箇所を抜き出して含めると、大いに役立ちます。
script/	スクリプト プラグインをインストールした場合は、このフォルダーにスクリプト ファイルを置きます。
filters.xml	ユーザー デフォルトのファイルフィルターです。
omegat.prefs	OmegaT の設定ファイルです。
segmentation.conf	ユーザー デフォルトの分節化規則です。
uiLayout.xml	GUI に関する情報（ウィンドウのレイアウト）が記録された XML ファイルです。

デフォルトの分節化規則とファイルフィルターよりも、プロジェクト専用の設定（前述）が存在した場合は、そちらが優先されます。個人設定ファイルの場所は、使用するプラットフォームによって異なります：

Windows 2000 および Windows XP	Documents and Settings\<User Name>\Application Data\OmegaT
Windows Vista および Windows 7	Users\<User Name>\AppData\Roaming\OmegaT
上記以外の Windows	<Something>\OmegaT（<Something> は Java が認識する「home」フォルダーの場所です）
Linux/Solaris/FreeBSD	<ユーザーホームフォルダー>/.omegat（.omegat はフォルダーで、先頭にドットがあるため、ls -a

	やそれに準じるコマンドでなければ一覧表示されません）
MAC OS X	<ユーザーホームフォルダー>/Library/Preferences/OmegaT
その他	<ユーザーホームフォルダー>

3. プログラムファイル

OmegaT は SourceForge サイトからダウンロードできるパッケージファイルとして提供されます。ここでは、標準的な Java 形式のプログラムを含んだプラットフォーム非依存のパッケージについて説明します。他のパッケージとしては、Linux 版の .tar パッケージ、Windows 版インストーラー（Java 実行環境ありまたはなし）、Mac OS X 版インストーラー、開発者向けソースコードなどがあります。

プラットフォーム非依存パッケージは、Java 実行環境（JRE）バージョン 1.6 が動くプラットフォームであれば、どこでも使用できます。そのプラットフォーム専用のパッケージがインストール済みの場合でも、共存できます。プラットフォーム非依存パッケージは圧縮ファイル（zip または TAR アーカイブ）で提供されているため、インストール先に選んだフォルダーに解凍する必要があります。たいていはダウンロードしたパッケージファイルをダブルクリックすれば、解凍できます。圧縮ファイルを解凍すると、以下の内容を含むフォルダーが生成されます：

ファイル/ フォルダー	内容
docs/	取扱説明書の全ファイルを格納します。取扱説明書に含まれる外部リンクへアクセスする場合は、これらのファイルをウェブブラウザーで開いてください。
images/	アイコンとロゴ画像を格納します。
lib/	OmegaT を利用するために必要な Java のファイルを格納します。
join.html	通常の HTML ファイルです。ウェブブラウザーで開くと、Yahoo! の OmegaT ユーザーグループへリダイレクトされます。参加は必須ではありませんが、もし参加すれば、OmegaT に関するディスカッションに加われるほか、別途準備されたファイルの入手や、不定期に行われる調査などに参加することが可能になります。過去ログは公開されており、閲覧するだけなら参加登録は必要ありません。
changes.txt	過去のバージョンから現在のバージョンまでの変更点の詳細が記されています。
license.txt	GNU GENERAL PUBLIC LICENSE（GNU 一般公衆利用許諾契約書）です。このライセンスにより、改変や再配布を含めた OmegaT の自由な利用が認められています。OmegaT の改変や再配布に興味がある場合は、それを実行する前にこの文書を慎重に読み、内容をよく理解しておいてください。不明点がある場合は、SourceForge サイト経由でメールを送るか、またはユーザーグループへ投稿して、プロジェクトのメンバーに問い合わせてください。
doc-license.txt	GNU GENERAL PUBLIC LICENSE（GNU 一般公衆利用許諾契約書）です。こちらは、取扱説明書に関するライセンスとなります。注意点は、上と同じです。
readme.txt	このファイルは、大変重要なファイルです。OmegaT を使用する前に必ず読み、内容を理解しておいてください。このファイルには、追加

	情報の入手先や、貢献方法など、OmegaT に関する全般的な内容が含まれています。このファイルは多くの言語に翻訳されています。
OmegaT	以下の 2 行からなるテキストファイルです： #!/bin/bash java java -jar OmegaT.jar $* Linux および Mac OS X のユーザーでしたら、役に立つことでしょう。OmegaT のプログラムフォルダーに移動してから、このファイルにコマンドラインで実行権限をつけてください（chmod +x OmegaT）。このファイルをコマンドラインから実行することで、OmegaT を起動できます。
OmegaT.bat	Windows の コマンドライン（コマンドプロンプト）から OmegaT を実行する際に使用するバッチファイルです。内容は以下の行のみです： java -jar OmegaT.jar %*
OmegaT.jar	主要な OmegaT のプログラムファイルです。OmegaT を起動するには、このファイルをコマンドライン上から実行するか、ファイルマネージャー上から（多くの場合そのアイコンをダブルクリックすることで）実行します。

第9章 翻訳対象ファイル

1. ファイル形式

OmegaT は翻訳対象として数多くのファイル形式に対応しています。ファイル形式は基本的に 2 種類に大別できます。プレーンテキスト形式と整形されたテキスト形式です。

1.1. プレーンテキストファイル

プレーンテキストファイルにはテキストのみが含まれます。したがってその翻訳は、単純に対訳を入力していくだけで済みます。OmegaT には、ファイルが文字化けしないよう、開く際のエンコーディングを指定する方法がいくつか用意されています。このようなファイルは、いわゆる「空白文字（ホワイトスペース）」以外には、整形用の情報を含んでいません。整形用の情報とは、テキスト揃えや段落、改ページなどの指定を指します。テキストの色や字体などの情報も、持ちません。現在の OmegaT は、プレーンテキストファイルのうち以下の形式に対応しています：

- ASCII テキスト（.txt など）
- エンコードされたテキスト
- Java リソースバンドル（*.properties）
- PO ファイル（*.po）
- INI ファイル（キー = 値 形式）（*.ini）
- DTD ファイル（*.DTD）
- DokuWiki ファイル（*.txt）
- SubRip 字幕ファイル（*.srt）
- Magento CE Locale CSV ファイル（*.csv）

他の形式のプレーンテキストファイルでも、OmegaT で扱うことができます。まずその拡張子を OmegaT が対応しているフィルター形式に追加します（たとえば .pod ファイルを「テキストファイル」フィルターに追加するなど）。次に、必要であれば適切な分節化規則を設定して前処理させればよいのです。

PO ファイルは、原文と訳文の両方を格納できるファイル形式です。したがって、PO ファイルはプレーンテキストファイルであると同時に、翻訳メモリでもあるといえます。原文分節に対応する訳文がまだ翻訳メモリ（project_save.tmx）に存在しなければ、翻訳内容はデフォルトの対訳（既定値訳文）として project_save.tmx に保存されます。もし原文分節に対応する別の訳文がすでに存在した場合、翻訳内容は（既定値訳文ではなく）「新しい訳文」として保存されます。

1.2. 整形されたテキストファイル

整形されたテキストファイルは、字体の種類や大きさ、色などの情報を含んでいます。ワードプロセッサーや HTML エディターで作成した文書に、よく見られます。この種のファイル形式は、文書整形に関する情報を保持するように設計されています。整形情報は、「文字を太字にする」ような単純なものから、異なる字体のサイズ、色、位置などが混在する表のような複雑なものまで、さまざまです。たいていの翻訳案件では、翻訳後の文書が原文と同じ見た目を保っているかどうかを重視します。OmegaT は整形情報を持つ文字列や単語に、扱いやすいタグを付けることで、この互換性を実現しています。原文テキストの整形を単純にしておくと、タグは減ります。もし可能であれば、文書中で使用する字体の種類や大きさ、色などをなるべく統一しておくと、翻訳作業を単純化でき、タグ関連のエラーを減らすことができます。OmegaT がファイルを処理する方法は、ファイル形式によって異なります。あるファイル形式特有の挙動は、ファイルフィルターの設定で管理できます。現在の OmegaT は、整形されたテキストファイルのうち以下の形式に対応しています：

- ODF - OASIS Open Document Format（*.ods、*.ots、*.odt、*.ott、*.odp、*.otp）
- Microsoft Office Open XML（*.docx、*.dotx、*.xlsx、*.xltx、*.pptx）
- (X)HTML（*.html、*.xhtml、*.xht）
- HTML ヘルプ コンパイラ（*.hhc、*.hhk）
- DocBook（*.xml）
- XLIFF（*.xlf、*.xliff、*.sdlxliff）-「原文 = 訳文」形式の一種
- QuarkXPress CopyFlowGold（*.tag、*.xtg）
- ResX ファイル（*.resx）
- Android リソース・ ファイル（*.xml）
- LaTex（*.tex、*.latex）
- Help（*.xml）とマニュアル（*.hmxp）ファイル
- Typo3 LocManager（*.xml）
- WiX ローカリゼーションファイル（*.wxl）
- Iceni Infix（*.xml）
- Flash の XML 出力（*.xml）
- Wordfast TXML ファイル（*.txml）
- Camtasia Studio for Windows（*.camproj）
- Microsoft Visio ファイル（*.vxd）

他の形式の整形されたテキストファイルでも、OmegaT で扱うことができます。その形式にふさわしい分節化規則を持つフィルターを探して、拡張子を追加すればよいのです。

2. その他のファイル形式

上記以外のプレーンテキストファイル、または整形されたテキストファイルの場合も、OmegaT で処理できる可能性があります。

外部ツールを使って、OmegaT が対応する形式に変換すればよいのです。訳文ファイルは最終的に、逆変換で原文ファイルの形式に戻す必要がある点に留意してください。たとえば拡張子に .doc がついた旧バージョンの Microsoft Word 形式ファイルは、既存のフィルターでは開けません。しかし、以下の手順で変換すれば、翻訳作業ができます：

- ファイルを ODF 形式に対応したプログラム（Apache OpenOffice や LibreOffice の Writer など）で開く
- ファイルを ODT 形式で保存しなおす
- ODT 形式ファイル（既存フィルター対応済み）を OmegaT で翻訳する
- 翻訳後に生成した訳文ファイルを、ODF 対応プログラムで開く
- ファイルを DOC 形式で保存しなおす

訳文ファイルの品質は、往復の変換工程によって左右されます。変換作業の前に、可能な限りのすべての設定を試すことをおすすめします。OmegaT のホームページ [http://www.omegat.org/]で、翻訳支援ツールの最新リストをチェックしてください。

3. 右から左へ表記する言語

原文と訳文の分節の行端寄せをどうするかは、プロジェクトの言語によって変わります。デフォルトでは、左から右へ表記する（LTR）言語は左寄せ、右から左へ表記する（RTL）言語は右寄せです。Shift+Ctrl+O（アルファベットの O です。数字の 0 ではありません）を押すと、この表示を切り替えることができます。Shift+Ctrl+O による切り替えは以下の 3 通りです：

- その言語でのデフォルトの行端寄せ
- 左寄せ
- 右寄せ

OmegaT で RTL モードを使用しても、作成した訳文ファイルの表示モードには何の影響も与えません。訳文ファイルの表示モードは、表示や編集に使用するプログラム（Microsoft Word のような）の中で修正する必要があります（詳細は、そのプログラムの取扱説明書を参照してください）。Shift+Ctrl+O を押すと、OmegaT 内のテキスト入力と表示の両方を、上記の 3 通りのいずれかに順に切り替えます。表示モードは、編集、参考訳文、用語集の 3 つのウィンドウそれぞれで個別に切り替えることができます。目的のウィンドウをクリックしてから、表示モードを切り替えてください。また、OmegaT にあるすべての入力フィールド（検索ウィンドウや分節化規則設定ダイアログなど）でも使用できます。

Mac OS X でも、同じ Shift+Ctrl+O を使用します（Cmd+Shift+O ではありません）。

3.1. 分節中での LTR（左から右）と RTL（右から左）文字列の混在

右から左へ表記する文章だけを入力している時は、デフォルトの（左から右への）表記が使用されます。しかし多くの場合、RTL 文章中に LTR 文章を混在させる必要が出てきます。たとえば、OmegaT が使用するタグ文字列、原文ファイルで LTR 表記された製品名、翻訳ファイル中のプレースホルダー、あるいは文章中の数値などです。このような場合、RTL（実際は両方向の）文章を適切に表示するため、RTL モードに切り替える必要があります。OmegaT を RTL モードにすると、原文と訳文の両方が RTL モードで表示される点に留意してください。仮に原文言語が LTR で訳文言語が RTL、またはその逆の場合を考えます。このとき、原文の確認、訳文の入力をそれぞれのモードでスムーズに行えるよう、RTL と LTR モードの切り替えを繰り返しながら作業する必要があるかもしれません。

3.2. RTL 分節における OmegaT タグ

上で述べたように、OmegaT が使用するタグは LTR 文字列です。RTL ⇔ LTR 言語間で翻訳作業を行うときは、原文ファイルからタグを正しく読み込み、それをまた訳文ファイルに適切に入力するため、翻訳作業者が RTL ⇔ LTR モードの切り替えを頻繁に行う必要があるかもしれません。

対象の文書にもよりますが、翻訳者に強くお勧めする点があります。原文の構成が許す限り、OmegaT 上に出てくるタグを減らすために、原文の見映えを単純化してください。「整形されたテキストを扱う」ページ末尾の「タグの取扱についてのヒント」を参照してください。起こりうる問題をなるべく見つけやすくするために、こまめにタグ検証（「タグ検証」を参照してください）を実行し、定期的に訳文生成（下記および 「メニュー」を参照してください）を実行してください。ヒント：翻訳作業はプレーンテキスト形式で行い、別のプログラムで必要な書式などを後から設定する方が、結果的には手間が省けるかもしれません。

3.3. 翻訳した RTL 文書の作成

訳文ファイルが生成されると、その表記方向は原文ファイルと同じになります。もし原文が LTR であった場合、訳文の表記方向は、表示するアプリケーション上で、手動で RTL に変更する必要があります。それぞれの出力形式によって、RTL 表記を取り扱う方法が決まります。詳細は、関連するプログラムの取扱説明書を参照してください。

.docx ファイルに関しては、以下の箇所が自動的に変更されます：

- 段落、セクション、表は bidi に設定されます
- Run 要素（テキスト要素）は RTL に設定されます

訳文ファイルを開くたびに表示設定を変えるという手間を避けるため、あらかじめ原文ファイルの表示設定を変えて保存しておく方法も検討してみましょう。この場合、訳文ファイルにとって適切な表示設定に合わせて、原文ファイルの表示を設定することになります。たとえば ODF ファイルはこのような変更が可能です。

第10章 翻訳入力行

［設定］→［翻訳入力行...］を選ぶと表示されるダイアログで、未翻訳の分節に移動した場合の翻訳入力行の初期値や、訳文の処理方法などを設定することができます：

図10.1 翻訳入力行の設定

翻訳入力行の設定

未翻訳の分節に移動したとき、その分節に挿入したい内容を以下から選択してください。

原文を挿入

原文を挿入しない

最も一致率の高い参考訳文を自動で挿入

しきい値となる一致率： 85

先頭に付加する文字列： [参考]

参考訳文挿入の場合は数値を変更する

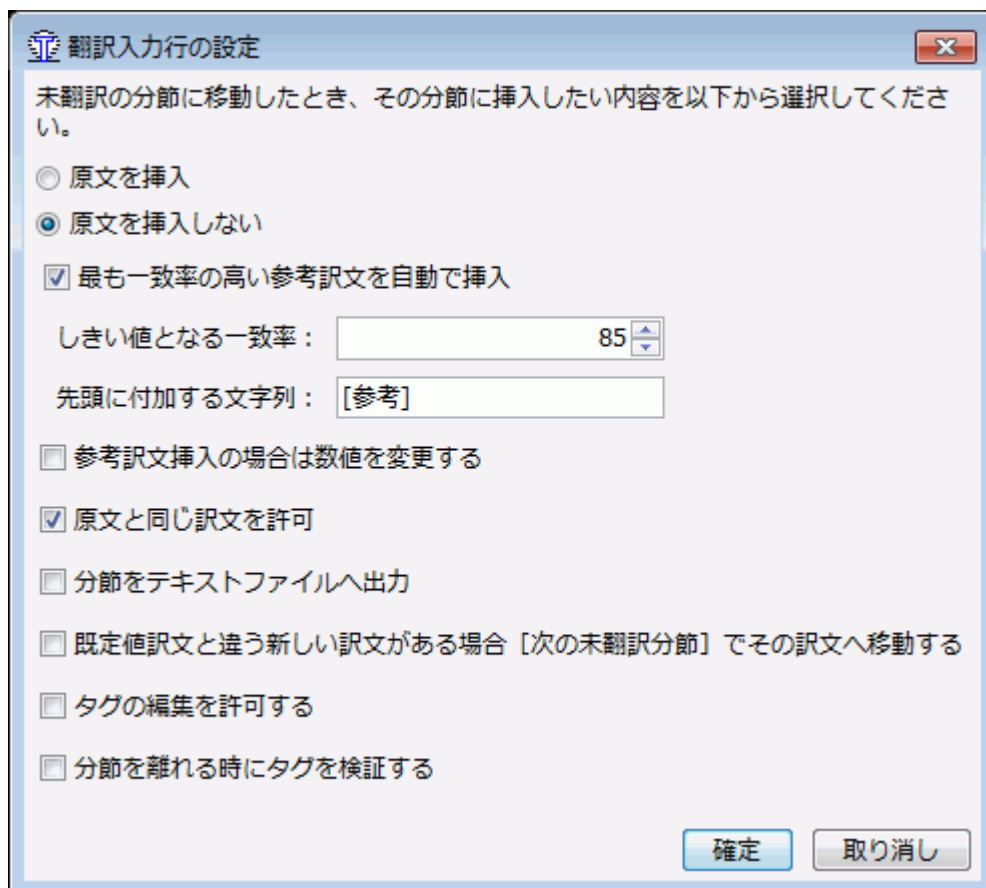
原文と同じ訳文を許可

分節をテキストファイルへ出力

既定値訳文と違う新しい訳文がある場合［次の未翻訳分節］でその訳文へ移動する

タグの編集を許可する

分節を離れる時にタグを検証する

確定　取り消し

翻訳作業は、分節間を行き来しながら、分節を編集することで進めていきます。分節を移動したときは、参考訳文を挿入することもできますし、原文のままにしておくこともできます。［設定］→［翻訳入力行...］から、以下の設定ができます：

原文を挿入
: 翻訳入力行に、対応する原文を入力した状態にします。商標や固有名詞など、あえて翻訳しない部分を多く含んでいるような文書の場合に非常に便利です。

原文を挿入しない
: OmegaT　は翻訳入力行を空欄にします。翻訳入力の前に、原文を削除するキー入力の手間（つまり Ctrl+A と Del）が省けます。この状態で、空訳文として残すこともできます。このような訳文は、未翻訳分節と区別するために、編集ウィンドウ上では <空訳文> と表示されます。空訳文に設定するには、分節を右クリックして「訳文を<空訳文>にする」を選びます。同じポップアップメニューに表示される「訳文を削除」という項目を選ぶと、現在の分節に入力済みの訳文を削除することもできます。これは Ctrl+A と Del キーの組み合わせで翻訳入力行をクリアするのと、同じことです。

最も一致率の高い参考訳文を自動で挿入
: OmegaT はその原文に最も近い参考訳文を自動挿入します。挿入するかどうかの一致率のしきい値（最小一致率）は、このダイアログで変更できます。先頭に付加する文字列（デフォルトでは何も付加しません）は、訳文が参考訳文の自動挿入によるものであることを示す目印に利用することができます。何らかの文字列（たとえば [参考] のような）を先頭に付加しておくと、参考訳文が適切かどうか後で確認する際に、自動挿入以外の訳文と区別しやすくなります。

ダイアログの下半分に並んでいるチェックボックスは、次の目的で用いられます：

参考訳文挿入の場合は数値を変更する
: この設定にチェックを入れると、参考訳文を挿入する際に（それが手動でも自動でも）、OmegaT は参考訳文に含まれる数値を、挿入先の原文に照らして変更しようとします。ただし、いくつかの制約があります：

- 編集中の原文分節と参考訳文は、同じ個数の数値を含んでいること
- 参考訳文とその原文に含まれる数値は、完全に一致していること
- 整数と単純な小数（小数点の記号としてピリオドを使うこと。たとえば 5.4 は小数とみなすが、5,4 や 54E-01 は小数とみなさない）だけを対象とする

原文と同じ訳文を許可
: 文書によっては、商標、名称、その他の固有名詞など、原文と同じ状態で残す必要のある箇所を含んでいる場合があります。変更すべきでないテキストを残しておくには 2 通りの方法があります。

まず、そのような箇所をまったく翻訳せずに残しておくことができます。OmegaT としては未翻訳の分節として残すことになります。こちらが初期設定です。もう 1 つの方法として、原文とまったく同じ訳文を入力することができます。OmegaT にはそこを翻訳したものとして記憶させることができます。そうする場合は、［設定］ →［翻訳入力行...］でで表示される画面の［原文と同じ訳文を許可］をチェックしてください。

分節をテキストファイルへ出力
: テキスト出力機能は、現在作業中の OmegaT プロジェクトのデータをプレーンテキストファイルに出力するものです。データは、分節に移動したタイミングで出力されます。ファイルは OmegaT の個人設定フォルダー配下の script フォルダーに作成されます。内容は、次の通りです：

- 編集中の分節の原文テキスト（source.txt）
- 編集中の分節の訳文テキスト（target.txt）
- Ctrl+Shift+C キーを押すか、またはメニュー［編集］→［選択領域を出力］を選んだ時点で選択されているテキスト（selection.txt）

ファイルの内容は、新しい分節に移動するタイミング（source.txt と target.txt）または新しい選択領域を出力するタイミング（selection.txt）で上書きされます。ファイル形式は、整形されないプレーンテキストファイルです。このプロセス全体を、Tck/Tcl ベースのスクリプトを介して操作· 制御することができます。詳しくは『OmegaT のテキスト出力機能を使う [http://www.omegat.org/en/howtos/text_export.html]』（英語）を参照してください。

既定値訳文と違う新しい訳文がある場合［次の未翻訳分節］でその訳文へ移動する
: たとえば同一の原文に対して、いくつかの異なる訳文が対応する場合があります。このような分節が存在していて、翻訳を慎重に進めたい場合は、このチェックボックスにチェックを入れてください。［次の未翻訳分節］で分節を移動するときに、翻訳済みか未翻訳かに関わらず、このような複数訳文に該当する分節で止まります。

タグの編集を許可する
: 編集作業中にタグの破損（たとえば部分的な削除）を防ぐには、このオプションのチェックを外しておいてください。タグ全体を削除したい場合は、このオプションにチェックを入れなくても、Ctrl+Backspace/Delete か、または全体を選択（Ctrl+Shift+Left/Right）して削除（Delete か Ctrl+X）で削除できます。

分節を離れる時にタグを検証する
: このオプションにチェックを入れると、原文と訳文のあいだでタグが同じか、分節を離れる時に検証して、もし違いがあれば警告します。

auto フォルダー由来の翻訳かどうかを記録する
: このオプションにチェックを入れると、auto フォルダー内の翻訳メモリから自動挿入された分節であることを示す情報が project_save.tmx ファイルに記録されます。その場合は、［表示］→［auto フォルダー由来の分節を色付け］にチェックを入れることで、該当する分節の背景色を変えることができます。

第11章 プレーンテキストを扱う

1. デフォルトエンコーディング

プレーンテキストファイル（拡張子 .txt の場合が多い）には、本文テキストしか含まれていません。どういった言語なのかをコンピューターに伝える方法は、明確に定義されていません。このような場合、OmegaT はテキストがコンピューターの言語設定と同じ言語で書かれているものとみなします。ファイルが 16 ビットでエンコードされたユニコード文字セットの場合は、特に問題は起きません。同じユニコードでも、8 ビットでエンコードされていた場合は、厄介な問題に直面します。たとえば次のような日本語のテキストがあった場合…

OmegaT とは、コンピューターを利用した翻訳ツールです。

… OS によっては、次のように表示されるかもしれません：

OmegaTВ∆ВЌБАГRГУГsГЕБ[Г^ВрЧШЧрВµВљЦ|ЦуГсБ[ГЛВ≈ВЈБВ

OmegaT を実行しているコンピューターがデフォルトの言語をロシア語に設定していた場合、テキストは日本語の文字ではなくキリル文字のアルファベットで表示されてしまいます。

2. OmegaT での解決方法

OmegaT には、主に 3 つの解決策があります。いずれも設定メニューにあるファイルフィルターを利用します。

ファイルのエンコーディングをユニコードに変更する	原文ファイルをテキストエディターで開き、エンコーディング UTF-8 で保存しなおします。このとき、拡張子を .txt から .utf8 に変更してください。OmegaT はこのファイルを UTF-8 ファイルとして自動的に認識します。これは最も理にかなった解決策で、あなたを厄介ごとから救ってくれます。
プレーンテキストファイルのエンコーディングを指定する	拡張子 .txt のファイルを例に取ります。ファイルフィルターダイアログからテキストファイルのフィルターを編集し、原文ファイルのエンコーディングを「<自動>」から対象の .txt ファイルに適したエンコーディングに変更します。たとえば上の例で言えば、日本語に対応したエンコーディングに変更します。
プレーンテキストファイルの拡張子を変更する	たとえば日本語のファイルであれば、拡張子を .txt から .jp に変えます。次に、ファイルフィルターダイアログからテキストファイルのフィルターを編集し、原文ファイル名の構成例を新しく追加します。例の場合は「*.jp」とします。エンコーディングには、適切な値を設定してください。

OmegaT は、プレーンテキストの取り扱いを簡単にするために、デフォルトで次の拡張子に対応しています。

- .txt ファイルは、エンコーディングを「<自動>」で認識します。これは、コンピューターのデフォルトのエンコーディングに合わせることを意味します。
- .txt1 ファイルは西ヨーロッパ言語のほとんどに対応する ISO-8859-1 エンコーディングとして解釈されます。
- .txt2 ファイルは中央、東ヨーロッパの言語のほとんどに対応する ISO-8859-2 エンコーディングとして解釈されます。
- .utf8 ファイルは OmegaT によって、（世界中ほぼすべての言語に対応する）UTF-8 エンコーディングとして解釈されます。

これらは［設定］メニューの［ファイルフィルター］一覧で［テキストファイル］を選択し、［編集...］を押すと［フィルターの編集］一覧で確認できます。たとえば、チェコ語で書かれたテキスト

ファイル（ほとんどの場合 ISO-9959-2 で書かれているでしょう）を扱う場合、拡張子を .txt から .txt2 に変更するたけで、OmegaT は内容を正しく解釈します。もちろん、安全のため、こうしたファイルはユニコード形式（ここでは .utf8 ファイル）への変換を検討してもいいでしょう。

第12章 整形されたテキストを扱う

通常、原文ファイルにある整形情報は、訳文ファイルにも同じく必要になります。対応しているフォーマット（具体的には DocBook、HTML、XHTML、Open Document Format（ODF）、Office Open XML（MS Office 2007 以降））のテキスト中にある整形情報は OmegaT のなかでタグとして表示されます。参考訳文を探すため異なる文字列間の類似度を調べる場合、通常、タグは考慮されません。翻訳した分節にあるタグは訳文ファイルにも含まれます。

1. 整形タグ

タグの名前：

タグ名は 1 から 3 文字の文字と数字からなります。タグにはショートカット文字が割り当てられますが、異なるタグが同じショートカット文字を共有する場合もあります。しかしタグの種類に応じて異なる数字が割り振られるため、区別できるようになっています。ショートカット文字は、できるだけタグの意味を反映するように選ばれます（たとえば太字の Bold には b、斜体の Italics には i など）。

タグの番号付け

タグに付けられる数字は、タググループごとに増えていきます。ここで言う「タググループ」とは、単一タグ（<i0> と </i0> のような）を指します。分節のなかで最初のグループ（ペアもしくは単一）が数字 0 となり、次が 1 となります。以下に示す最初の例には、3 つのタググループ（ペア、単一、そしてまた別のペア）があります。2 つめの例にあるのは、1 つのグループ（ペア）だけです。

単一とペア：

タグは常に単一であるか、ペアになっています。単一タグは、タグの周囲にあるテキストには影響しない整形情報を示します（たとえば、追加の空白や改行などです）。

<b0><Ctr+N></b0>, <br1><b2><Enter></b2><分節 2132>

<br1>が単一タグで、周囲のテキストにはまったく影響しません。通常、ペアになったタグは、開きタグと閉じタグの間に挟まれた文字列に適用される整形情報を示しています。下の例の <b0> と </b0> がペアになっており、テキスト「log.txt」に作用しています。開きタグは必ず、対応する閉じタグの前に現れる点に、注意してください。

稼動監視ログファイル（<b0>log.txt</b0>）とエラー。<分節 3167>

OmegaT では、文の分節化が行われる前にタグを作成します。そのため、分節化規則によってはペアのタグが、連続する 2 つの分節にまたがって分かれることがあります。これは必ずしもタグの破損を意味しませんが、タグ検証機能は大事を取って、分節を両方ともエラーとして報告します。

2. タグ操作

タグを操作する場合には細心の注意が必要です。もし誤って変更してしまうと、最終的なファイルの整形が壊れてしまうかもしれません。基本原則は「タグが連続する場合は、その順番を崩さない」です。しかし、ある一定のルールを遵守すれば、この基本原則を曲げることも可能です。

タグの複製：

タググループを複製するには、単に好きな場所にコピーするだけです。タグがペアになっているタググループでは、開きタグは閉じタグの前になければならないということを忘れないでください。グループを複製した箇所は、元と同じように整形されます。

例：

<b0>この書式</b0>を、こちらにも適用しましょう。<分節 0001>

複製後：

<b0>この書式</b0>を、<b0>こちらにも適用</b0>しました。<分節 0001>

タググループの削除：

タググループを削除するには、分節から取り除けばいいだけです。ペアになっているタググループでは、整形情報を適切に削除するため、開きタグと閉じタグの両方を削除しなければならないことに注意してください。さもなければ訳文ファイルが壊れてしまう可能性があります。タググループを削除することで、関連する整形情報が訳文ファイルから削除されます。

例：

<b0>この書式</b0>を削除します。<分節 0001>

削除後：

この書式を削除しました。<分節 0001>

3. タググループの入れ子

タググループの順序を変更することで、タググループが他のタググループのなかに入れ子になることがあります。この場合、外側のタググループが内側のタググループを完全に囲む限り、問題ありません。言いかえると、ペアになったタグを動かす場合、開きタグと閉じタグの両方が他のタググループのなかに収まるか、あるいは外に出ていればよいのです。そうでない場合は、翻訳ファイルが壊れてしまい、開かなくなる危険性もあります。

例：

<b0>書式設定</b0><b1> 1 番</b1>を、書式設定 0 番の入れ子にしましょう。<分節 0001>

入れ子になった後：

<b0>書式設定<b1> 1 番</b1></b0>は、書式設定 0 番の入れ子になりました。<分節 0001>

4. タググループの重なり

誤った操作の結果、複数のタグペアで囲まれた箇所が（入れ子ではなく）一部重なってしまうと確実に翻訳ファイルの整形が壊れてしまい、場合によっては翻訳ファイルがまったく開かなります。

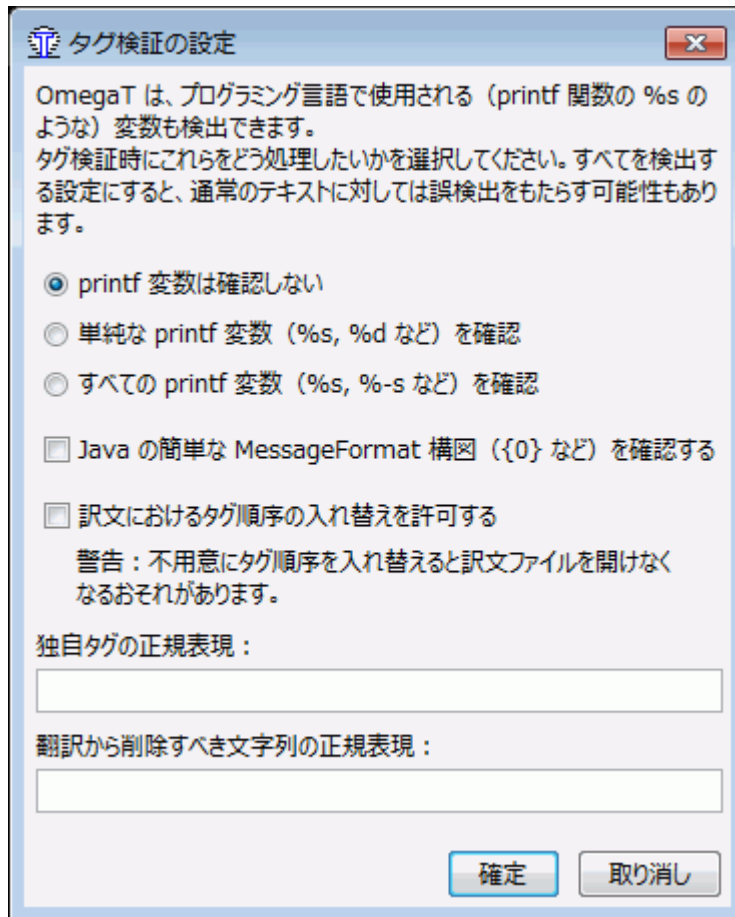
例：

<b0>書式設定</b0><b1> 1 番</b1>は、大変なことになります！<分節 0001>

誤った操作後：

<b0>書式設定<b1> 1 番</b0></b1>は、大変なことになりました！<分節 0001>

5. タグ検証の設定

タグの取り扱いをカスタマイズするには、メニューの [設定] → [タグ検証...] を選んでください。表示されるダイアログ上で、いくつかの規則を設定できます。

そこでカスタマイズした設定は、すべての原文ファイルに適用されます。整形されたテキストのような、一部のファイル形式にだけ適用されるわけではありません。

- printf 変数 - 確認しない、単純なものだけ確認、すべて確認

  OmegaT は、プログラミング言語で使用される（たとえば %s のような）変数（書式化文字列）も検出できます。検出レベルは、何も確認しない、単純な printf 変数（%s, %d など）を確認、すべての変数を確認、の３つから選択できます。

- Java の簡単な MessageFormat 構図を確認する

  このチェックボックスにチェックを入れると、OmegaT は単純な Java の MessageFormat タグ（{0} など）を認識するようになります。

- 独自タグの正規表現

  独自にタグとして認識させたい文字列の正規表現を、ここに入力します。タグ検証機能は、独自タグについても、OmegaT のタグと同じように、個数と順番が変わっていないか確認します。

- 翻訳から削除すべき文字列の正規表現

  訳文ファイルに含めたくない文字列を、正規表現で入力します。その文字列が訳文分節に存在した場合は、気づきやすいように該当箇所が赤字で表示されます。参考訳文の検索時には、削除すべき文字列は無視されます。もし削除文字列を含む参考訳文と含まない参考訳文とがあった場合は、削除文字列を含む参考訳文に対して５％の固定ペナルティが与えられます。このような場合に、削除した文字列を含む参考訳文の類似度が１００％になることは、ありません。

6. タググループの検証

タグ検証機能は、（故意か過失かによらず）変更されたタグを検知し、影響のある分節を表示します。Ctrl+Shift+V　でこの機能を使用できます。専用のウィンドウには、壊れたもしくは誤ったと思しき

タグを含む分節がすべて表示されます。タグ検証機能により、タグの修正と翻訳文書の再生成が簡単になります。Ctrl+Shift+V により表示されるウインドウには 3 列の表があります。それぞれ分節へのリンク、原文の分節、訳文の分節です。

図12.1 タグ検証項目

1	A different display font can be selected via the <b0>Display Font</b0> dialog. Open it via the <i1>Settings</i1>><i2>Display Font...</i2> menu item. The font type and size can be changed from the dialog.	表示用の字体は、<b0>表示用字体</b0>ダイアログから変更できます。ダイアログを開くには、<i1>［設定］</i1>→［表示用字体...］メニュー項目を選んでください。字体の種類と大きさは、そのダイアログで変更できます。

原文と訳文で比較しやすいよう、タグは青い太字で強調表示されます。分節番号をクリックすると、編集ウィンドウにその分節を表示できます。必要に応じて訳文を修正（上の例でいえばタグ <i2></i2> ペアの欠落）します。Ctrl+Shift+V を押すと、他のエラーを修正するためにタグ検証ウィンドウに戻ることができます。タグエラーは、訳文分節にある一連のタグの順序や個数が、原文の分節のそれと一致しない場合に発生します。タグ操作には、必要なもの、行っても問題ないものもありますが、中には翻訳後の文書を作成した際に問題となるものもあります。

7. タグ管理についてのヒント

原文テキストを単純にする

タグがあるということは、多くの場合、ある種のフォーマット・整形がそこで行われているということを表します。原文の整形を単純にしておくと、タグは減ります。もし可能であれば、文書中で使用する字体の種類、大きさ、色などをなるべく統一しておくと、翻訳作業を単純化でき、タグ関連のエラーを減らすことができます。タグに対してどういった操作が可能か、タグ操作のセクションを見てください。もし、タグの取り扱いが面倒な場合や整形が翻訳内容にそれほど影響しないときは、訳文にタグを付けないことが、もっとも簡単な解決策になりうることを忘れないでください。

ペアになったタグに特に注意する

原文のタグを確認する必要はあるが、翻訳後の文書に整形情報を含む必要があまりないのであれば、訳文にタグを入れる必要はありません。この場合、ペアになっているタグには特に注意を払ってください。片方のタグだけ削除してしまうと文書の整形情報が壊れてしまうからです。タグはテキストの中に含まれているので、分節化規則を使ってタグの少ない分節を生成することが可能です。これは比較的高度な機能であり、うまく使いこなすにはあるていどの経験が必要です。

OmegaT は今のところ、整形に関する誤りをすべて自動で見つけだすことはできません。また、それが間違いなのか、訳文言語にあうよう意図的に整形情報を改善したものなのか、区別することもできないでしょう。したがって、タグのエラーが見過ごされた時には、生成した訳文ファイルが何かおかしかったり、最悪の場合は、ファイルを開くことができなくなるおそれもあります。

第13章 翻訳メモリ

1. OmegaT の翻訳メモリ

1.1. tmx フォルダー – 場所と目的

OmegaT のプロジェクトは、翻訳メモリファイル（拡張子 tmx）を 5 箇所に配置します。

omegat フォルダー

omegat フォルダーには、project_save.tmx と、おそらく複数のバックアップファイルが含まれています。project_save.tmx ファイルには、プロジェクトを開始してから記録されたすべての分節が含まれています。このファイルはプロジェクトに必ず存在します。内容は、常に原文分節のアルファベット順に並べ替えされています。

メインプロジェクトフォルダー

メインプロジェクトフォルダーには 3 つの TMX ファイルが作成されます。プロジェクト名-omegat.tmx、プロジェクト名-level1.tmx、プロジェクト名-level2.tmx です。「プロジェクト名」の部分には、実際のプロジェクト名が入ります。

- level1 ファイルには、文書情報のみが含まれます。
- level2 ファイルには、OmegaT 特有のタグが適切な TMX タグに変換された状態で含まれます。したがって整形情報は、TMX レベル 2 の翻訳メモリに対応した他の翻訳ツールや、OmegaT 自身で使用することができます。
- OmegaT ファイルには、OmegaT 特有の整形情報を持ったタグが含まれるので、他の OmegaT プロジェクトで使用できます。

これらのファイルは、プロジェクトの内部翻訳メモリ project_save.tmx の複製と言えます。ただし、現在の原文に存在しない分節は含みません。これらのファイルは名前が変わっただけで、その内容については同一性を保っています。そのため、別のプロジェクトの tm フォルダーにコピーして使うことができます（次の項を参照してください）。

tm フォルダー

tm フォルダーには補助的な翻訳メモリ（TMX ファイル）をいくつでも置いておくことができます。これらのファイルは、上記で説明した 3 つの形態のいずれでも使用できます。他のコンピューター翻訳支援ツールがエクスポート（あるいはインポート）できる TMX ファイルの形式も、たいていはこの 3 つのうちのどれかです。可能であれば、OmegaT 専用形式の TMX ファイル（上記を参照）を使うのが、もちろん望ましい方法です。そうすれば、分節内の整形情報も利用できます。

tm フォルダーに配置された翻訳メモリの内容は、参考訳文の候補を探す際に参照されます。これらのファイルに記録された翻訳済みの分節情報は、現在の原文分節とじゅうぶん一致した場合、参考訳文ウィンドウに表示されます。

また補助的な翻訳メモリの中の原文分節と、翻訳中のテキストとが同一の場合の挙動を、設定 → 翻訳入力行... ダイアログで設定できます。たとえば、デフォルト設定のままであれば、補助的な翻訳メモリから自動挿入された訳文には、先頭に「[参考]」という接頭辞が付きます。これにより、翻訳者が後から翻訳を見直すときに、その訳文が参考訳文のままなのかどうかを簡単に識別できます（「翻訳入力行」の章を参照してください）。

tm フォルダーの翻訳メモリに、同一の原文分節が複数存在し、しかもそれぞれの訳文が異なる、という事態はじゅうぶんに起こりえることです。複数の TMX ファイルは、そのファイル名の順番で読み込まれます。また TMX ファイル内の分節は、一行ずつ順次読み込まれます。この場合、原

	文分節が一致する最後の分節が、優先的に使われることになります（注意：もちろん、そもそもこのような事態が起こらないようにできるなら、それが一番望ましい対応です）。 tm フォルダーに配置する TMX ファイルは、gzip 形式で圧縮しておくこともできます。
tm/auto フォルダー	補助的な翻訳メモリの内容がすべて正しい、ということがあらかじめはっきりと分かっている場合は、そのファイルを tm/auto フォルダーに置いてください。そこから自動挿入される訳文には、「[参考]」という接頭辞が付きません。この方法は、効率の良い下訳になります。「auto」の翻訳メモリから原文が一致する分節が見つかった箇所は、ユーザーの手を一切わずらわせることなく、すべてプロジェクトのメイン翻訳メモリに移し替えられます。
tm/enforce フォルダー	OmegaT の project_save.tmx よりも優先度が高いことが明らかな TMX がある場合は、そのファイルを /tm/enforce に配置してください。そこからの分節は、既存の翻訳を無条件に上書きします。
tm/mt フォルダー	mt と名付けられたフォルダー内の TMX から編集ウィンドウに参考訳文を挿入すると、その分節は一時的に赤い背景色に変わります。その分節から離れると、背景色は通常に戻ります。
tm/penalty-xxx フォルダー	品質の高い翻訳メモリと、そうではない翻訳メモリを区別できると役に立つ場合があります。分野の違いや依頼主の都合、改版の状態など、理由はいろいろあるでしょう。「penalty-xxx」（「xxx」は 0 から 100 までの数値）という名前のフォルダーに翻訳メモリを置くと、そこから見つかった参考訳文の一致率は、フォルダー名の末尾に付いた数値の分だけ引き下げられます。たとえば「Penalty-30」という名前のフォルダーがあったとします。そこに置かれた翻訳メモリから 100% 一致する分節が見つかった場合、一致率は 70% に引き下げられて表示されます。このペナルティは、参考訳文候補に併記される 3 種類の一致率すべてに適用されます。一致率 75、80、90 の場合は、45、50、60 に引き下げられます。

必要に応じて、プロジェクト内の翻訳可能な全分節を含む TMX ファイル（OmegaT 形式）を、指定した場所に作成することもできます。下記「4. 疑似翻訳メモリ」の項を参照してください。

すべての翻訳メモリはプロジェクトを開いた時点でメモリ上に読み込まれます。プロジェクトのメイン翻訳メモリについては、バックアップが定期的に作成されます（次章を参照）。そして project_save.tmx もまた、プロジェクトを閉じるか再読み込みした時点で保存または更新されます。したがって、翻訳作業の途中で補助的な翻訳メモリをフォルダーに追加した場合でも、それを使うために作業中のプロジェクトをいったん終了する必要はありません。単純にプロジェクトを再読み込みしてください。そうすれば、追加した翻訳メモリを使えるようになります。

あるプロジェクトのために多種多様な翻訳メモリをどこに配置するかは、ユーザーに任されています（「プロジェクトの設定」のプロジェクト設定ダイアログを参照してください）。

状況に応じて、いろいろな作業方針が考えられます。たとえば：

同じ案件に対して複数のプロジェクトがある場合：プロジェクトの構造を共通化し、原文フォルダーと訳文フォルダーのみを個別化します（Source = source/order1、target = target/order1 など）。order1 の分節は、order2 などの他の関連プロジェクトには存在しません。そのため、メインの翻訳メモリを order1 から引き継いだ場合、それらは現在の原文に存在しない分節として扱われることになるでしょう。それでも、依然として参考訳文としては有効です。

同じ案件を複数の翻訳者で分担する場合：原文ファイルを source/Alice、source/Bob のように分割し、それをチームメンバー（Alice、Bob など）に配分します。メンバーは各自でプロジェクトを作成し、作業が終わるか所定の段階まで達したら、自分の project_save.tmx を送ります。project_save.tmx ファイルが集まったら、用語や表記揺れなどの不整合を直します。こうして新しいバージョンのマスター翻訳メモリが作成され、メンバー各自の tm/auto フォルダーに配置されるか、または project_save.tmx ファイルと置き換えることになります。訳文フォルダーの構造は、チーム共通で問題ありません。こうしておくと、プロジェクト全体の訳文ファイルが問題ないか、いつでもチェックできます。

1.2. TMX ファイルのバックアップ

OmegaT は、翻訳内容をプロジェクト内の omegat/project_save.tmx ファイルに随時蓄積していきます。

また、プロジェクトを開くか再読み込みするたびに、同じフォルダーに project_save.tmx.YYYYMMDDHHNN.bak という名前で翻訳メモリのバックアップファイルを保存します。「YYYYMMDDHHNN」の部分は、前回保存した日付を 4 桁の年（YYYY）、月（MM）、日（DD）、時（HH）、分（NN）で表します。

もし翻訳中のデータを失ってしまったと気づいた場合、以下の手順に従ってください：

1. プロジェクトを閉じる
2. 現在の project_save.tmx ファイルの名前を一時的に変更する（たとえば project_save.tmx.temporary）
3. 探している内容（直近だったり、前日の最終だったり）にもっとも近い翻訳メモリのバックアップファイルを選択する
4. そのファイルをコピーして、名前を project_save.tmx に変更する
5. プロジェクトを再度開く

1.3. TMX ファイルと言語

TMX ファイルには、ある同じ分節に対し、複数の異なる言語で表記された翻訳単位が含まれています。翻訳単位は、少なくとも 2 つの翻訳単位言語要素（tuv）から構成されています。ある言語要素は、原文にも訳文にもなりえます。

プロジェクトの設定によって、どの言語要素が原文で、どの言語要素が訳文かが決まります。OmegaT は、プロジェクトの原文言語と訳文言語に対応する翻訳単位言語要素を参照し、それぞれ原文分節と訳文分節として使用します。OmegaT は言語コードの指定方法として、次の 2 種類の標準規約を認識します：

- 2 文字の言語コード（例：日本語は JA）、または
- 2 文字か 3 文字の言語コードに続けて 2 文字の国コード（例：EN-US - 代表的な言語コードと国コードの一覧については「付録A 言語設定 - ISO 639 コード一覧 」を参照）

プロジェクトの言語コードと TMX の言語コードが完全に一致すると、分節がメモリに読み込まれます。言語コードが一致して国コードが一致しない場合でも、分節はやはり読み込まれます。言語コードと国コードいずれも一致しない場合、その分節は無視されます。

TMX ファイルには通常、複数の異なる言語要素から成る翻訳単位が含まれています。もし原文分節と一致する言語要素が見つかったにもかかわらず、訳文言語に当たる言語要素が存在しない場合は、他のすべての言語要素が（その言語が何であるかに関わらず）読み込まれます。たとえばプロジェクトの言語ペアが DE-FR だったとしましょう。この場合、翻訳メモリに DE-FR ペアが見つからず、DE-EN しかなかったとしても、それを表示することによって何がしかの助けにはなるはずです。

1.4. 現在の原文に存在しない分節

project_save.tmx ファイルには、プロジェクトを開始してから翻訳したすべての分節が含まれています。プロジェクトの分節化規則を変更したり原文ファイルを削除したりすると、一部の候補が現在の原文に存在しない文字列として参考訳文ウィンドウに表示されることがあります。このような参考訳文は、すでに存在しないが、修正前にその翻訳が記録された分節を参照しています。

2. 翻訳メモリの再利用

最初にプロジェクトが作成された時点では、メインの翻訳メモリ project_save.tmx は空の状態です。この翻訳メモリは、翻訳作業を通じて徐々に蓄積していきます。このプロセスを加速させるために、既存の翻訳を再利用することができます。ある文が、すでに過去に一度翻訳されていて、しかもその内容が

正しい場合、それをもう一度翻訳し直す必要はありません。翻訳メモリはまた、参照翻訳も含むことができます。欧州連合の多言語法律文書などはその典型です。

OmegaT プロジェクト内で訳文ファイルを生成すると、プロジェクトの翻訳メモリは、そのプロジェクトフォルダーに 3 種類のファイルとして出力されます（前述の説明を参照）。これら 3 つの TMX ファイル（-omegat.tmx、-level1.tmx、-level2.tmx）は「エクスポートした翻訳メモリ」、つまり現在のプロジェクトの内容を 2 言語で外部出力したものと、みなすことができます。

過去のプロジェクトで使用した翻訳メモリを再利用したいとき（たとえば、今度作成する新しいプロジェクトが「過去のものと非常に似ている」または「以前に使用したかもしれない専門用語を含んでいる」というような場合など）、この翻訳メモリを「入力用翻訳メモリ」として、新しいプロジェクトにインポートできます。この場合は、使いたい翻訳メモリを新しいプロジェクトの tm フォルダーか tm/auto フォルダーに配置してください。前者では、翻訳メモリから見つかった分節が参考訳文ウィンドウに表示されます。後者では、下訳として訳文分節に自動挿入されます。

デフォルトでは、tm フォルダーはプロジェクトのルートフォルダーの下（たとえば MyProject/tm）にあります。しかし必要であれば、プロジェクトの設定でその位置を変えてもかまいません。これは、過去に作成した翻訳メモリを何度も使用するような場合に便利です。たとえば、同じ顧客向けであったり、同じ分野の翻訳であったりする場合が挙げられます。この場合、使いやすい手順は以下の通りです：

- 特定の顧客や分野の翻訳メモリ用として、ハードディスク内に 1 つフォルダー（たとえば「保管フォルダー」）を作成する
- プロジェクトを閉じて終了したとき、常に「エクスポートした」3 つの翻訳メモリファイルを、プロジェクトのルートフォルダーから、その保管フォルダーにコピーしておく
- 同じ顧客または分野について、プロジェクトを新しく作成するときは、［プロジェクト］→「プロジェクト設定...］ウィンドウで、［翻訳メモリフォルダー］にその保管フォルダーを指定する

プロジェクトを開くときに、tm フォルダー内の全 TMX ファイルが読み込まれます。持っているすべての TMX ファイルをそこに置いておくと、OmegaT の動作が不必要に遅くなることがあるので、注意してください。それらの内容がいったん現在のプロジェクトの project_save.tmx に蓄積されたなら、もう必要ないファイルを削除してもよいでしょう。

2.1. 翻訳メモリのインポートとエクスポート

OmegaT は TMX バージョン 1.1 から 1.4b（レベル 1 と レベル 2 の両方）をインポートすることができます。これにより、他のツールで作成された翻訳メモリを、OmegaT で読み込むことができます。しかし、OmegaT はレベル 2 の（文書情報だけでなく、整形情報も含む）TMX ファイルに完全には対応していません。それでも、レベル 2 の TMX ファイルをインポートして、テキスト情報を OmegaT で利用することは可能です。ただし参考訳文の一致率は多少下がるかもしれません。

OmegaT が翻訳メモリファイル（TMX ファイル）を読み込む際には、非常に厳密な処理が行われます。ファイルに何かのエラーを見つけると、OmegaT はその場所を通知します。

一部で、不正な TMX ファイルを生成してしまうツールがあることが確認されています。そのような翻訳メモリを OmegaT に参照させるには、事前にファイルを修復しておく必要があります。でなければ、OmegaT はエラーでそのファイルを読み込めないでしょう。修正は簡単な操作で行えることが多く、OmegaT が出力するエラーメッセージはじゅうぶんその助けになります。不明な点があればユーザーグループに問い合わせてください。

OmegaT はバージョン 1.4 の TMX ファイル（レベル 1、レベル 2 とも）を出力します。出力されたレベル 2 TMX ファイルは、レベル 2 の仕様に完全に準拠してはいませんが、それに近い水準です。TMX レベル 2 に対応した他の翻訳ツールは、このファイルから参考訳文を問題なく利用できます。もし、テキスト情報のみが必要な場合（整形情報が必要でない場合）は、OmegaT が作成したレベル 1 ファイルの方を使用してください。

2.2. 特定文書向け翻訳メモリの作成

翻訳者のあいだで、TMX ファイルを共有したい場合があるとします。しかし、特定の文書に関する分節を除外したいか、逆にそれだけを含めたいという状況です。ProjectName-omegat.tmx そのものを共

有するのは、論外です。以下に示す解決法は、いろいろあるやり方のひとつにすぎません。しかし試すのはとても簡単で、ファイルを破損する危険性もありません。

- まず、まったく新しくプロジェクトをひとつ作成します。共有したい TMX に合わせて、言語ペアやプロジェクト名を適切に設定します（このあと作成される TMX ファイルには、このプロジェクトの名前が付きます）。
- 翻訳メモリの対象にしたい原文文書を、source フォルダーにコピーします。
- 元の（上記文書の翻訳を含む）翻訳メモリを、新しいプロジェクトの tm/auto フォルダーにコピーします。
- 新しいプロジェクトを開始します。タグのエラーが無いか、Ctrl+T で確認してください。未翻訳分節は Ctrl+U で確認します。問題が無ければ、Ctrl+D を押して訳文ファイルを生成し、その内容を確認します。
- プロジェクトを終了しましょう。メインのプロジェクトフォルダーに 3 つの TMX ファイル（上記参照）が出力されているはずです。これには、指定した言語ペアの、かつ source フォルダーにコピーした文書だけの翻訳が含まれています。あとで使うために、それらを安全な場所にコピーしておきましょう。
- このプロジェクトを再利用するのは、避けたほうが賢明です。他と取り違えかねないので、用が済んだらプロジェクトごと削除するか、あるいはアーカイブ化して作業スペースから追い出しましょう。

2.3. 翻訳メモリの共有

翻訳者のチームで作業する場合、配布された専用のプロジェクトで各自が孤立して作業するよりも、ひとつの翻訳メモリを共有して作業できるほうが望ましいでしょう。

OmegaT は SVN と Git という、2 つの一般的なバージョン管理システム（VCS）に対応しています。これらは、オープンソースのライセンス下で利用可能です。OmegaT の場合は、すべてのプロジェクトフォルダーが VCS の管理下に入ります。原文ファイルフォルダーやプロジェクト設定のみならず、翻訳メモリも同様です。詳しくは次の章を参照してください。

2.4. TMX ファイルと代替言語ペア

ここに、完了したプロジェクト（たとえばオランダ語の原文を英語に翻訳したもの）があるとします。次に、これを中国語に翻訳する必要が発生しました。翻訳者はオランダ語を解しませんが、英語であれば理解できます。このような状況では、蘭中翻訳を、蘭英の翻訳メモリを介して行うしかありません。

この例における解決策は、既存の翻訳メモリを tm/tmx2source/ フォルダーにコピーし、訳文言語を示すために ZH_CN.tmx とリネームすることです。編集ウィンドウの現在分節には、オランダ語の原文分節と共に英語の対訳も表示されます。翻訳者は両方を見ながら、中国語の訳文を作成することができます。

重要：補助用の TMX ファイルは必ず「XX_YY.tmx」形式にリネームしなければなりません。「XX_YY」はプロジェクトの訳文言語と同じコードです。たとえば上の例では、「ZH_CN.tmx」となっています。プロジェクトの原文言語コードと TMX の原文言語コードは、同一でなければなりません。例では「NL」がそれです。特定の言語ペアについて関連付けできるのは、TMX ファイルひとつだけです。したがって、関連付けたい翻訳メモリが複数ある場合は、それらを XX_YY.tmx にマージする必要があります。

3. 対訳付きの原文ファイル

ある種の原文ファイル（PO ファイルや TTX ファイルなど）は二ヶ国語形式、すなわち原文と翻訳メモリを兼ねた状態になっています。この場合、原文ファイルに付属する対訳は、project_save.tmx に移行されます。翻訳メモリ内に他に一致する分節がなければ、移行された翻訳は、デフォルトの翻訳になります。翻訳メモリ内に他の一致する分節が記録済みの場合、移行された翻訳は、複数訳文の「新しい訳文」の方として扱われます。複数訳文になる場合、翻訳が「既定値訳文」になるか「新しい訳文」になるかは、原文分節の読み込まれる順番に依存します。

原文ファイルに付属する対訳情報はすべて、コメントウィンドウと参考訳文ウィンドウにも表示されます。PO ファイルの場合は、複数訳文の「新しい訳文」扱いになった対訳情報に対して 20% のペナルティが課せられます（たとえば 100% 一致しても、80% に減点されます）。また原文分節に「[参考]」と表示されます。

分節化された TTX ファイルを読み込む際は、［設定］→［翻訳入力行...］の［原文と同じ訳文を許可］にチェックが入っていれば、原文＝訳文の状態で表示されます。この状態は混乱を招きやすいので、TTX ファイルの場合はチェックを外しておく方が良いかもしれません。

4. 疑似（Pseudo）翻訳メモリ

注記

興味のある上級者のユーザーのみを対象としています。

翻訳を開始する前に、分節に対する前処理や、OmegaT 上での作業とは別の処理を加えたい場合があるかもしれません。たとえば、テストのために疑似的な翻訳を行いたいような場合です。OmegaT はプロジェクトの全分節を含む、別の TMX ファイルを作成することができます。この TMX ファイル中の翻訳内容は以下のいずれかです：

- 翻訳（訳文）が原文とまったく同じであるもの（既定の設定）
- 翻訳（訳文）が空のもの

この TMX ファイルには任意の名称を付けることができます。疑似翻訳メモリは、次のコマンドラインで作成できます：

java -jar omegat.jar --pseudotranslatetmx=<ファイル名> [--pseudotranslatetype=[equal|empty]]

<ファイル名> には、作成したいファイル名を指定します。絶対パスか、あるいは OmegaT の起動フォルダーを基準にした相対パスで指定できます。２つ目の引数 --pseudotranslatetype は省略可能です。設定値は equal（既定値、原文＝訳文）または empty（訳文分節が空）です。作成した TMX ファイルは、任意のツールで処理することができます。これを OmegaT で流用する場合は、ファイル名を project_save.tmx に変更し、プロジェクトの omegat フォルダーに置いてください。

5. 翻訳メモリの更新

ごく初期のバージョンの OmegaT は、段落単位でしか原文を分節化できず、また HTML と Open Document ファイルに対しては、整形用のタグに対する番号付けを適切に行えない場合がありました。OmegaT では、参考訳文の品質を向上し、すでに存在する翻訳内容をより活用するために、このような古い形式の TMX ファイルを見つけると、その場で新しい形式に更新します。そのために手作業で何かを行う必要はありません。

プロジェクトのメインの TMX ファイルがバージョンアップされるのは１度だけです。最新の形式で project-save.tmx に書き込まれます。それ以外の以前の TMX ファイルは、プロジェクトが読み込まれるたびにその場で更新されます。OmegaT ではファイルフィルターに変更が加えられたため、分節化の内容がまったく異なる可能性があります。ごくまれに、翻訳メモリを手動で更新する必要があるかもしれませんので、注意してください。

第14章 原文の分節化

翻訳メモリツールは、分節と呼ばれるテキスト単位で処理を行います。OmegaT がテキストを分節化する方法は 2 つあります。段落単位の分節化と、文単位の分節化（「ルールベースの分節化」とも呼ばれます）です。分節化の方法を選択するには、メインメニューから［プロジェクト］→［プロジェクト設定...］を開き、表示されるチェックボックスを使用してください。段落単位の分節化は、高度に文学的または芸術的な翻訳のような、一部のケースでは有用です。これらの翻訳では、文脈に応じて翻訳者が文の順番を入れ替えることがあるためです。しかし大半のプロジェクトでは、文単位の分節化が望ましいでしょう。その方が、過去の翻訳と一致しやすい（再利用しやすい）ためです。文単位の分節化を選択した場合、分節化規則は、メインメニューの［設定］→［分節化...］で設定できます。

信頼できる分節化規則が多くの言語で設定済みのため、新たな独自規則を作成する必要は、あまりないでしょう。一方で、特定の状況では、翻訳する原文に合わせて分節化規則を調整することで、大きく生産性を上げることができるようになります。

警告：ファイルフィルターの設定を変更すると、分節が変化することがあります。場合によっては、翻訳を一からやり直すことになるかもしれません。このとき、以前は有効だった翻訳済みの分節は、翻訳メモリ内で、現在の原文に存在しない分節になります。またプロジェクトを開いているときに分節化の設定を変更すると、変更を適用するためにプロジェクトの再読み込みが必要になります。

OmegaT は下記の通りに一連の処理を実施します：

構造単位の分節化
: OmegaT は、まず構造単位の分節化を行うため、文書の構文を解析します。この処理のあいだ、分節作成のために参照される内容は、原文ファイルの構造のみです。

  たとえば、テキストファイルの場合は、改行で分節化するか、空行で分節化するか、または、まったく分節化しないかのいずれかです。整形情報を持ったファイル（ODF 文書、HTML 文書など）は、ブロックレベル（段落）のタグを基準に分節化されます。XHTML や HTML ファイルでは、オブジェクトの属性も翻訳の対象になることがあります。このような場合は、属性値だけを分節として取り出します。

文単位の分節化
: 原文ファイルを論理的な区切りで分節化した後、OmegaT はさらにそれらを文単位で分節化します。

1. 分節化規則

分節化の流れは以下のように示すことができます：テキスト中を 1 文字ずつ動くカーソルをイメージしてください。それぞれのカーソル位置について、規則を適用できるかどうか、決められた順序で確認します。すなわち、カーソル位置の左側にあるテキスト全体が、前方の正規表現に合致するかどうか。合致した場合はさらに、カーソル位置の右側にあるテキストが、後方の正規表現に合致するかどうか。適用できる規則が見つかれば、そのカーソル位置についてはそれ以上規則を確認しません。見つかった規則が例外（分割しない）の場合は、何もせずにカーソルを次へ進めます。分節（分割する）の場合は、カーソル位置で改節してから、カーソルを次へ進めます。

規則には、次に示す 2 種類があります：

分節規則
: 原文テキストを 2 つの分節に分割します。たとえば「Did it make sense? I was not sure.」というテキストは、意味的に 2 つの分節に分けられるべきです。そのためには「?」に関する分節規則（後ろにスペースと大文字で始まる単語が続く場合、という条件付きで）があるほうがよいでしょう。改節するための規則を定義する場合は、［分割する/しない］チェックボックスにチェックを入れます。

例外規則
: 分割すべき「でない」テキストを指定します。「Mrs. Dalloway」というテキストはピリオドを含みますが、2 つに分節化されるべきではありません。したがって、ピリオドの前に「Mrs」（他に Mr、Dr、prof など）がある場合についての例外規則が必要になります。改節を禁止するための規則を定義する場合は、［分割する/しない］チェックボックスにチェックを入れずにおきます。

ほとんどのヨーロッパ言語と日本語については、定義済みの分節規則で十分なはずです。翻訳しようとする言語に対して、より意味のあるわかりやすい分節化を行うために、さらに柔軟に例外規則を追加してもよいでしょう。

2. 規則の優先順位

言語の構成例について一致する分節化規則は、すべて与えられた優先順で適用されます。したがって、特定の言語のために設定した規則は、デフォルトのものより高い優先順位を与えるほうがよいでしょう。たとえば、カナダフランス語（FR-CA）用の規則は、フランス語（FR.*）用の規則よりも優先順位を高くし、なおかつデフォルト（.*）の規則よりも高くしておきます。こうしておけば、カナダフランス語を原文とするプロジェクトでは、まずカナダフランス語用の規則、次にフランス語用の規則、最後にデフォルトの規則、という順序で適用されます。

3. 規則の新規作成

分節化規則を大きく変更することは、特に最初の下訳が終わった後では、避けるべきです。しかし略称を認識させるために規則を追加するような、限定的な変更であれば、それは改善に値します。

規則集を編集するには、［分節化規則の設定］ウィンドウ上部にある表から、目的の項目をクリックします。するとウィンドウ下部に、選択した規則集の内容が表示されます。

新しい言語コードの正規表現を追加するには、上部の表の横にある［追加］をクリックしてください。表の最下部に新しい行が作成されます（確認のために下までスクロールさせる必要があるでしょう）。言語名と、言語コードの正規表現を変更してください（言語コードの一覧は付録A 言語設定 - ISO 639 コード一覧 を参照してください）。定義する言語コードの文法は正規表現の文法に従います。作成する規則群が、言語-国ペアに対するものである場合、［上へ］ボタンを使って、先頭に移動させておくことをおすすめします。

前方の正規表現と後方の正規表現を追加する追加する構成例が文法的に正しいか、また意図する対象にちゃんと適用されるか確認するためには、正規表現をその場でテストできるような支援ツールの使用をお勧めします。詳しくは「正規表現」の章を参照してください。手始めに、既存の規則を参考に真似してみるのも良いでしょう。

4. 簡単な設定例

目的	前方の正規表現	後方の正規表現	備考
ピリオド（「.」）の後ろと、それに続くスペースやタブの前で分節を作る	\.	\s	正規表現「\.」は、文字「.」を表します。正規表現「\s」は任意の空白文字（スペース、タブ、改行など）を表します。
「Mr.」の後では分節化しない	Mr\.	\s	これは例外規則です。［分割する/しない］チェックボックスには、チェックを入れずにおきます。
「。」（日本語の句点）の後ろで分節化する	。		［後方の正規表現］は空欄としておいてください。
「M.」、「Mr.」、「Mrs.」、「Ms.」の後ろでは分節化しない	Mr??s??\.	\s	例外規則です。? 文字の使用方法については「正規表現」の章を参照してください。

第15章 検索

1. 検索ウィンドウ

Ctrl+F で文字列検索ウィンドウを開き、［検索文字列］欄に検索したい単語または語句を入力します。

または、編集ウィンドウであらかじめ検索したい単語や語句を選択してから、Ctrl+Fを押します。この場合、［検索文字列］欄に単語または語句が自動的に入力されます。検索ウィンドウはいくつも同時に開いておくことができます。しかし、デスクトップを散らかさないように、検索が終わったら閉じることを忘れないでください。

［検索文字列］欄のドロップダウンボタンをクリックすると、直近の検索履歴が 10 個まで表示されます。

2. ワイルドカードの使用

完全一致検索とキーワード検索では、ワイルドカードとして「*」と「?」を使用できます。Word ユーザーにはおなじみでしょうが、意味は次の通りです：

- 「*」は、0 個以上の任意の文字と一致します。たとえば、検索文字列が「run*」である場合は、「run」「runs」「running」に一致します。
- 「?」は任意の 1 文字と一致します。たとえば、検索文字列が「run?」である場合は、「runs」や、「running」の中の「runn」の部分に一致します。

一致した文字列は、青い太字で表示されます。「*」と「?」は正規表現において特別な意味を持つことに注意してください。ここで説明したワイルドカード文字は、あくまで完全一致検索とキーワード検索にだけ当てはまるものです（下記を参照してください）。

3. 検索方法と設定

ラジオボタンで検索方法を選択してください。以下の検索方法を利用できます：

完全一致検索
: 指定した文字列をそのまま含む分節を検索します。完全一致検索は、指定された文字列を「語句」として検索します。つまり、複数の単語を指定した場合、それらの単語の並びまで同じ語句を含む検索結果のみが表示されます。文字列「open file」を検索した場合、「open file」とは一致しますが、「file opened」や「open input file」とは一致しません。

キーワード検索
: 指定したすべてのキーワードを含む分節を検索します。指定した順序は問いません。複数の単語の登場順を問わないような検索を行いたい場合は、それらの単語を空白で区切って入力し、［キーワード検索］を選択してください。OmegaT は、指定した単語をすべて含む、すべての分節の一覧を表示します。キーワード検索は、Google などのインターネット検索エンジンにおける「すべてを含む」検索（AND 検索）に似ています。キーワード「open file」を検索した場合、「open file」と一致するのはもちろんですが、「file opened」や「open input file」「file may not be safe to open」などとも同じように一致します。

正規表現
: 検索文字列が正規表現として扱われます。たとえば上の図で指定されている「[a-zA-Z]+[öäüqwß]」という検索文字列は、ドイツ語配列のキーボードでは不自然な文字が使われている箇所を訳文から検索しています。正規表現は、文字列を検索するにあたって、とても効果的な方法です。詳しくは「正規表現」の章を参照してください。

上記いずれの方法についても、以下のオプションを指定できます：

- 大文字と小文字を区別：指定した文字列そのものを検索します。大文字と小文字の違いも含めて、一致を判断します。
- スペース文字で nbsp も検索：検索文字列に入力されたスペース文字を、通常のスペース文字だけでなく、ノーブレークスペース（\u00A）にも一致させます。

- 原文を検索：原文分節から検索します。
- 訳文を検索：訳文分節から検索します。
- メモを検索：メモから検索します。
- コメントを検索：コメントから検索します。
- 全分節：翻訳済み分節と未翻訳分節の両方から検索します。
- 翻訳済み分節：翻訳済み分節だけから検索します。
- 未翻訳分節：未翻訳分節だけから検索します。
- 表示：繰り返し分節：内容が同じ分節が複数見つかった場合でも、検索結果にすべての分節を繰り返し表示します。
- 表示：ファイル名：分節ごとに、その分節が属するファイル名を先頭に表示します。
- プロジェクトを検索：内部メモリにチェックを入れると、プロジェクト専用の翻訳メモリ（project_save.tmx ファイル）から検索します。翻訳メモリにチェックを入れると、tm フォルダー内の翻訳メモリファイルが検索対象になります。用語集にチェックを入れると、glossary フォルダー内の用語集ファイルが検索対象になります。
- ファイルから検索：単独のファイル、またはフォルダーに含まれるファイル群を検索します。（翻訳メモリではない）複数のファイルを検索する場合、対象となるファイルは、OmegaT が対応している原文ファイル形式に限定されます。そのため、OmegaT は TMX ファイルの検索にも対応していますが、ファイル検索機能からは対象外になります。

［詳細設定］ボタンをクリックすると、上の図でも見ることができるように、追加の条件欄（翻訳者の名前や、翻訳が最後に更新された日時）が表示されます。

4. 検索結果の表示

［検索文字列］欄に文字列を入力し［検索］ボタンをクリックすると、プロジェクトから、その文字列を含むすべての分節が表示されます。OmegaT は、同一の分節を 1 つの実体として扱うので、見つかった最初の分節のみを表示します。検索された分節は、プロジェクト中で現れる順に表示されます。翻訳済みの分節は、その原文とともに表示されます。上段が原文、下段が訳文です。未翻訳の分節は、原文のみが表示されます。

結果表示欄の分節をダブルクリックすると、修正できるように、編集ウィンドウ上で該当する分節に移動します。必要であれば、用語の修正を行うため、検索ウィンドウに戻って、検索された次の分節を確認することもできます。

同時に複数の検索ウィンドウを開くことができます。検索ウィンドウのウィンドウタイトルには検索文字列が設定されるため、検索ウィンドウが複数開いていても、容易に識別できます。

5. 編集ウィンドウの分節に検索フィルターをかける

検索結果の分節間を移動しやすいように、編集ウィンドウに対して検索フィルターをかけることができます。最下部にある［フィルター］ボタンをクリックすると、編集ウィンドウには現在の検索条件に一致する分節のみが表示されます。分節間の移動には、通常の移動コマンド（［次の分節］や［次の未翻訳分節］など）を使用できます。

注意事項：

- 検索結果には上限（既定で 1,000 件）があるため、一般的な語句を検索した場合、編集ウィンドウには検索条件に合致するすべての分節が表示されるのではなく、その中の 1,000 件だけが表示されることになります。
- 検索条件に合致する分節を含むファイルが見つからない場合は、空白の画面が表示されます。

- ［繰り返し結果も表示］の設定がオフになっている場合、繰り返しに該当する分節は編集ウィンドウ上でも表示されません。

フィルターを解除するには、［フィルターを解除］ボタンをクリックするか、またはプロジェクトを再読み込みしてください。

第16章 置換を実行

1. 検索ウィンドウ

Ctrl+K で検索と置換ウィンドウを開き、［検索文字列］欄に検索したい語句または正規表現を入力します。

［検索］ボタンをクリックすると、検索条件に一致したすべての分節が表示されます。

［置換後の文字列］欄に新しい語句を入力し（正規表現は使えません）、以下のいずれかをクリックします：

- すべて置換：すべての検索文字列を置換後の文字列で置き換えます（事前に置換件数を示した確認ダイアログが表示されます）。
- 置換フィルター：編集ウィンドウに置換用のフィルターをかけ、編集ウィンドウのヘッダーに現れるボタンを使って、1 件ずつ置換します。［置換して次へ］か［次を検索］をクリックしていき、最後に［フィルターを解除］でフィルターを解除します。
- 閉じる：変更なしにウィンドウを閉じます。

1.1. 検索オプション

検索オプションは、検索ウィンドウに表示される検索オプションとほぼ同じです。

例外的に、未翻訳分節にチェックを入れると、翻訳前の分節の原文も検索対象になります。

検索条件に一致した未翻訳分節に対しては、原文を訳文にコピーした上で置換が実行されます。置換が実行されなかった場合、未翻訳分節はそのままです。つまり、原文は訳文にコピーされません。

第17章 正規表現

文字列検索と分節化規則で使用する正規表現は、Java がサポートしているものです。より詳細な情報については Java の正規表現に関する技術文書 [http://docs.oracle.com/javase/jp/6/api/java/util/regex/Pattern.html]を参照してください。以下の構文一覧と使用例も参照してください。

注記

この章は上級者のユーザーを対象にしています。独自の分節化規則を定義したり、こみいった条件での検索を必要とする場合に、参照してください。

表17.1 正規表現 - フラグ

構文	一致対象
(?i)	大文字と小文字を区別しない一致を有効にします（デフォルトでは大文字と小文字を区別します）

表17.2 正規表現 - 文字

構文	一致対象
x	文字 x（以下に示すもの以外）
\uhhhh	16 進値 0xhhhh を持つ文字
\t	タブ文字（'\u0009'）
\n	改行（ラインフィード、LF）文字（'\u000A'）
\r	キャリッジリターン（CR）文字（'\u000D'）
\f	用紙送り文字（'\u000C'）
\a	警告 (ベル) 文字（'\u0007'）
\e	エスケープ文字（'\u001B'）
\cx	x に対応する制御文字
\0n	8 進値 0n を持つ文字（0 <= n <= 7）
\0nn	8 進値 0nn を持つ文字（0 <= n <= 7）
\0mnn	8 進値 0mnn を持つ文字（0 <= m <= 3、0 <= n <= 7）
\xhh	16 進値 0xhh を持つ文字

表17.3 正規表現 - 引用

構文	一致対象
\	次の文字をエスケープします。以下のメタ文字（!$()*+.<>?[\]^{|}）自体を一致対象としたい場合に入力が必要です
\\	たとえば、この構文ではバックスラッシュ文字を表します
\Q	\E までのすべての文字をエスケープします
\E	\Q で開始された引用をエスケープします

表17.4 正規表現 - Unicode ブロックとカテゴリのクラス

構文	一致対象
\p{InGreek}	ギリシャ語ブロックの文字（単純 ブロック [http://docs.oracle.com/javase/jp/6/api/java/util/regex/Pattern.html#ubc]）

構文	一致対象
\p{Lu}	大文字（単純 カテゴリ [http://docs.oracle.com/javase/jp/6/api/java/util/regex/Pattern.html#ubc]）
\p{Sc}	通貨記号
\P{InGreek}	ギリシャ語ブロック以外の文字（否定）
[\p{L}&&[^\p{Lu}]]	大文字以外の文字（減算）

表17.5 正規表現 - 文字クラス

構文	一致対象
[abc]	a、b または c（単純クラス）
[^abc]	a、b、c 以外の文字（否定）
[a-zA-Z]	a 〜 z または A から Z（範囲）

表17.6 正規表現 - 定義済みの文字クラス

構文	一致対象
.	任意の文字（行末記号を除く）
\d	数字：[0-9]
\D	数字以外：[^0-9]
\s	空白文字：[\t\n\x0B\f\r]
\S	非空白文字：[^\s]
\w	単語構成文字：[a-zA-Z_0-9]
\W	非単語文字：[^\w]

表17.7 正規表現 - 境界正規表現エンジン

構文	一致対象
^	行の先頭
$	行の末尾
\b	単語境界
\B	非単語境界

表17.8 正規表現 - 最長一致数量子

構文	一致対象
X?	X、1 または 0 回
X*	X、0 回以上
X+	X、1 回以上

注記

最長一致数量子は、指定した文字となるべく多い文字数で一致します。たとえば a+? は文字列 aaabbb にある aaa と一致します。

表17.9 正規表現 - 最短一致数量子

構文	一致対象
X??	X、1 または 0 回
X*?	X、0 回以上
X+?	X、1 回以上

注記

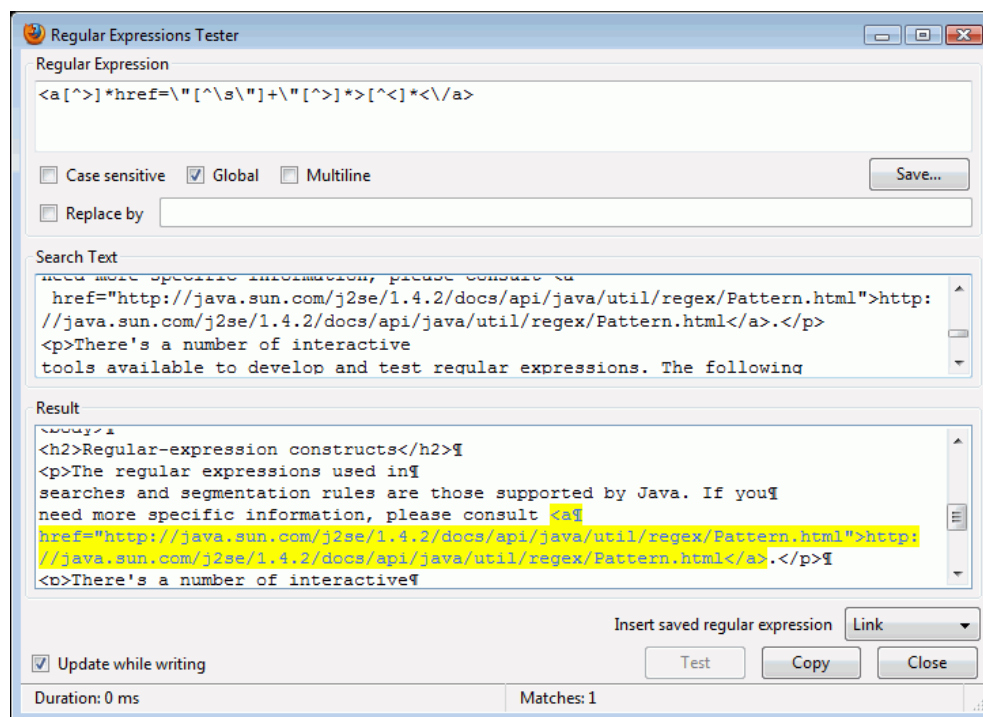
最短一致数量子は、指定した文字となるべく少ない文字数で一致します。たとえば a+? は文字列 aaabbb にある最初の a と一致します。

表17.10 正規表現 - 論理演算子

構文	一致対象
XY	X の直後に Y
X|Y	X または Y
(XY)	XY を 1 つのグループとして扱います

1. 正規表現関連ツールとその使用例

正規表現の作成やテストを行える対話式のツールは、たくさんあります。それらのツールは、たいてい同じ使い方ができます（実例は、下の Regular Expression Tester を参照してください）。入力した正規表現（図では最上段の「Regular Expression」）にしたがって対象テキスト（中段の「Search Text」）を検索し、その結果（下段の「Result」）を表示します。

図17.1 Regex Tester

Windows、Linux、FreeBSD 向けの単体ツールを探しているなら、The Regex Coach [http://weitz.de/regex-coach/] を確認してみてください。このツールは、上の例（Firefox アドオン）とほとんど同じように使えます。

正規表現の便利な構文例は、OmegaT 自体にも含まれています（［設定］→［分節化規則...］を参照してください）。以下は、特に翻訳メモリを検索する場合に便利な正規表現の一覧です：

表17.11 正規表現 - 翻訳で使える正規表現例

正規表現	検索対象
(\b\w+\b)\s\1\b	重複している単語
[\.,]\s*[\.,]+	カンマまたはピリオドの後ろに、0 個以上のスペースが続いて、再びカンマまたはピリオド

正規表現	検索対象
\. \s+$	行末で、ピリオドの後ろに余分なスペースがある箇所
\s+a\s+[aeiou]	英語：母音で始まる単語の前に「an」でなく「a」がある箇所
\s+an\s+[^aeiou]	英語：子音で始まる単語の前に「a」でなく「an」がある箇所
\s{2,}	2 個以上の空白
\.[A-Z]	ピリオドの直後に大文字が続く箇所 - 新しい文章の前の空白が欠落している可能性あり？
\bis\b	「is」を検索（「this」や「isn't」などに一致させない）

第18章 辞書

1. 辞書のダウンロードとインストール方法

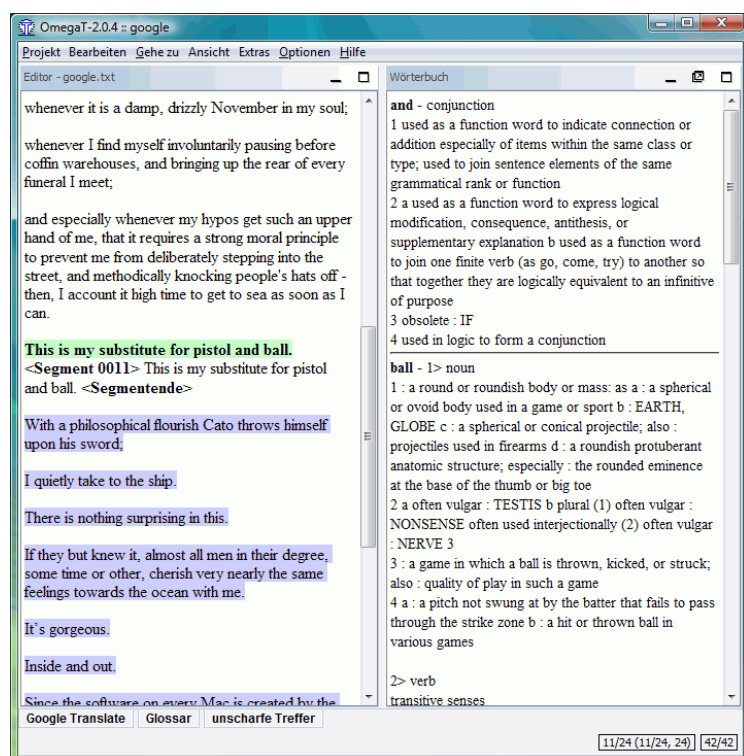
OmegaT は、StarDict 形式および Lingvo DSL 形式の辞書に対応しています。StarDict 形式の辞書は、次の手順でインストールしてください：

- 必要な言語の組み合わせの辞書を探します。辞書を提供している場所には、たとえば StarDict プラットフォーム [http://stardict.sourceforge.net/index.php]があります。
- 辞書ファイルをダウンロードします。これは TAR アーカイブ（拡張子 tar.bz または tar.bz2）である必要があります。
- untar コマンド（Windows であれば、たとえば WinRAR などのユーティリティ）で、そのファイルを展開します。展開先は、プロジェクトフォルダー配下の「dictionary」フォルダーの中です。展開すると、拡張子が dz、idx 、ifo の 3 つのファイルができるはずです。

利用できるのは、対訳形式の辞書に限りません。この辞書機能を使って、多種多様な情報を参照できます。たとえば：

- Webster's Revised Unabridged Dictionary (1913)
- Longman Dictionary of Contemporary English
- The Britannica Concise Encyclopedia
- …などです。

辞書の中には文書が何も付属していないものがありますが、これは自由に使用できます。一方、上記に挙げたような辞書は GPL ライセンスに従います。以下は、Merriam Webster 10th dictionary を実際に参照している様子です：

図18.1 Merriam Webster 10th dictionary - 使用中の様子

2. 辞書に関するトラブルシューティング

- プロジェクト設定で指定したフォルダー（もしくはそのサブフォルダー）中に、辞書ファイルが存在しますか？メニュー［プロジェクト］→［プロジェクト設定...］(Ctrl+E)で、辞書フォルダーの場所を確認できます。
- そのフォルダーには同じ名前のファイルが３つ（拡張子違いで）存在していますか？
　もし１つしか見当たらなければ、拡張子を確認してみましょう。tar.bz　か　tar.bz2だったら、圧縮ファイルを展開し忘れています。

第19章 用語集 (Glossaries)

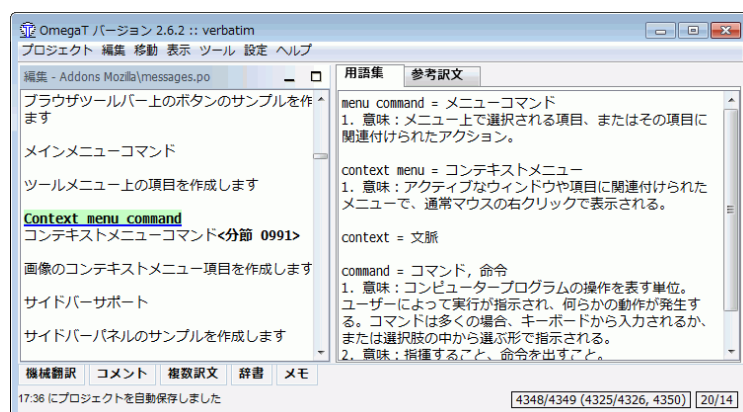
用語集は OmegaT から参照用として使用できるファイルで、作成と更新は手動で行います。

ある OmegaT のプロジェクトに 1 つ以上の用語集がある場合、現在の分節中に見つかった用語が、用語集ウィンドウに自動的に表示されます。

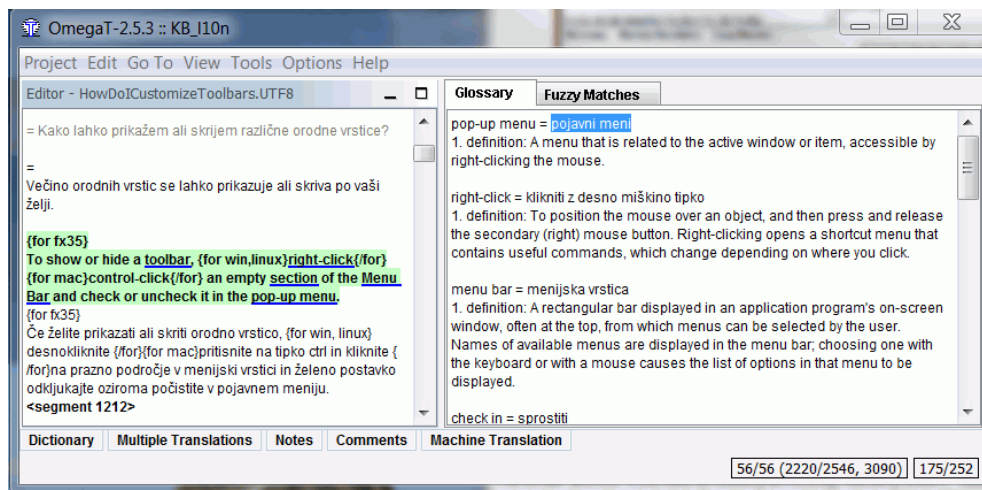
用語集ファイルの場所と名前は、プロジェクト設定ダイアログで設定できます。拡張子は必ず .txt か .utf8 にしてください（拡張子を指定しなければ、自動的に付加されます）。ファイルの場所は glossary フォルダー内を指定する必要があります。ただし、直下ではなくもっと深い階層に置いてもかまいません（たとえば glossary/sub/glossary.txt）。場所を指定した時点では、ファイルが必ずしも存在する必要はありません。用語を追加するときに、（必要であれば）自動的に作成されます。ファイルがすでに存在した場合、その形式と文字コードに問題が無ければ、特にメッセージは表示されません。新しい用語は常にタブ区切り形式の文字コード UTF-8 で追加されます。このとき、既存の内容は変更されません。したがって、既存のファイルへの影響は最小限に留まります。

1. 使い方

すでにある用語集を使用する場合は、プロジェクトを作成後、glossary フォルダーにそのファイルを置いてください。OmegaT はプロジェクトを開いた際、自動的にそのファイルを検出します。用語集のファイルにある用語が現在の分節中に見つかれば、用語集ウィンドウに表示されます：

図19.1 用語集ウィンドウ

「=（等号）」の前にあるものが原文の用語です。「=」の後ろに対応した訳文が表示されます。見出し語ごとにコメントを追加できます（上の図の「意味：」で始まる行がコメントです）用語集が機能するのは、用語集の項目と完全一致する部分のみです（その活用形などは検索されません）。（訳注：トークナイザープラグインを使用すると、改善される場合があります。詳細は付録 A を参照してください）新しい用語は、翻訳中でも手動で（テキストエディターなどを別途用いて）用語集ファイルに追加することができます。追加した用語は、用語集ファイルの変更を保存するまでは反映されません。

用語は、必ずしも 1 つの単語である必要はありません。次の例を見てください。

図19.2 用語集における複数単語による項目 - 例

下線が引かれた項目「pop-up menu」は、用語集ウィンドウに「pojavni menu」の見出しで表示されています。用語集ウィンドウでテキストを選択し、その選択範囲内を右クリックすると、訳文分節のカーソル（キャレット）位置に選択テキストが挿入されます。[1]

2. ファイル形式

用語集ファイルは単純なテキストファイルで、タブ区切りの 3 つの列（カラム）からなるリストの形をしています。最初の列と 2 番目の列はそれぞれ原文、訳文の用語が入ります。 3 番目の列には追加情報を入れることができます。訳文用の列は、空にしておくこともできます。つまり、原文の用語とコメントのみを含む登録も可能です。

用語集ファイルの文字エンコードには、システムのデフォルト（拡張子を .tab とします）または UTF-8（拡張子を .utf8 または .txt とします）または UTF-16LE（拡張子を .txt とします）のいずれかを使用できます。また CSV 形式にも対応しています（日本語訳註：こちらの文字エンコードは UTF-8 のみの対応です）。この形式は、タブ区切りと同じように、原文の用語、訳文の用語、そしてコメントの列から構成されます。ただし列は、カンマ記号「,」で区切られます。また文字列は引用符「"」で括られます。そのため、列区切りと区別して、文字列内にカンマを含むことができます。

"This is a source term, which contains a comma","これは訳文の用語です, カンマも含みます"

プレーンテキストファイルだけでなく、TBX 形式にも対応しています（読み取り専用）。.tbx ファイルの場所は glossary フォルダー内を指定する必要があります。ただし、直下ではなくもっと深い階層に置いてもかまいません（たとえば glossary/sub/MyGlossary.tbx）。

TBX（Term Base eXchange）は、オープンな XML ベースの標準規格で、構造化された用語集データの互換を目的としています。TBX は国際標準規格として、LISA と ISO によって承認されています。もし何か用語を扱うシステムを使っているなら、用語集データを TBX 形式で出力できる機能を備えている可能性が十分にあります。マイクロソフト用語集 [https://www.microsoft.com/Language/ja-JP/Terminology.aspx] は 100 種類近くの言語でダウンロード提供されており、基本的な IT 用語集として使うことができます。

注意：MultiTerm が出力する .tbx 形式ファイルは、信頼性に劣っているようです（2013 年 11 月時点）。MultiTerm の場合は、.tab 形式ファイルの方で出力することをお勧めします。

3. 用語集の作成手順

プロジェクト設定では、書き込み可能な用語集ファイルの名前を入力することができます（本章の最初を参照してください）。用語集ウィンドウを右クリックするか、Ctrl+Shift+G を押して、新しい用語を

[1]このとき、もし訳文が語形変化のある言語（図の場合はスロベニア語）だった場合、作業はこれで終わりません。挿入する用語「pojavni menu」は主格の表現ですが、これをたとえば所格の「pojavnem meniju」などに変える必要があるかもしれないのです。だから時には、用語集やそのショートカットを使うよりも、最初から正しい用語を手入力した方が早い場合もあるでしょう。

追加します。ダイアログが開くので、そこで原文の用語、対応する訳文、任意のコメントを入力してください。

用語集ファイルの内容はプロジェクトを開いたときと再読み込み時に読み込まれ、メモリ上に保持されます。用語集ファイルの更新は、とても簡単です。Ctrl+Shift+G を押して、新しい用語、その訳文、任意のコメントを入力（入力欄は TAB キーで移動できます）し、保存するだけです。用語集ウィンドウの内容は、すぐに更新されます。

書き込み可能な用語集ファイルの場所は、［プロジェクト］→［プロジェクトの設定...］で表示されるダイアログで設定できます。有効な拡張子は、.txt と .utf8 です。

注意：もちろん、タブ区切り形式の単純なファイルを作るには、他にもいろいろなやり方があります。Windows 上で Notepad++ を使ったり、Linux 上で GEdit を使ったり、表計算ソフトを使うことだって、あるでしょう。UTF-8（または UTF-16LE）を正しく扱えて、空白文字（ホワイトスペース）を表示できる（TAB 記号を見落とさずに済みます）プログラムなら、何を使っても構いません。

4. 用語集の優先順位

優先度の高い用語集（デフォルトでは glossary/glossary.txt）から見つかった結果は、用語集ウィンドウと用語ヒントの最初に表示されます。

優先度の高い用語集とそれ以外の用語集からの結果が混在することがあります。優先度の高い用語集から見つかった用語は、太字で表示されます。

5. TRADOS Multiterm の使用

TRADOS Multiterm のテキスト出力機能を用いて作成した用語集は、拡張子を .tab として、最初のフィールドを原文の用語、２番目を訳文の用語としておけば、特に修正を加えることなく OmegaT で使用することができます。システムオプションで［タブ区切り出力］を用いた場合は、残したい２つの項目の前にある、先頭の５つの列（Seq. Nr、Date created など）を削除する必要があります。

6. 用語集に関するよくある問題

問題：用語が何も表示されません。

- glossary フォルダーに用語集ファイルがない
- 用語集ファイルが空である
- 項目がタブ文字で区切られていない

- 用語集ファイルが有効な拡張子（.tab、.utf8、.txt）ではない
- 原文の項目と「完全に」一致する用語が用語集にない - たとえばその項目が複数形である場合
- 用語集ファイルが有効な文字エンコーディングで保存されていない
- 現在の分節に、用語集と一致する用語が１つもない
- 以上のうち１つまたは複数の原因を解決したものの、プロジェクトを再読み込みしていない

問題：用語集ウィンドウの中で、いくつかの文字が適切に表示されない（しかし同じ文字が編集ウィンドウには適切に表示されている）

- 拡張子とファイルの文字エンコーディングが一致していない

第20章 TaaS の使用

1. 概要

TaaS はヨーロッパ言語（およびロシア語）向けの用語管理サービスです。https://demo.taas-project.eu/info で提供されています。公開と非公開、両方のデータを取り扱うことができます。既存の文書から用語を抽出し、複数の情報源から訳語を自動生成することで、非公開の用語集を作成できます。

2. 公開用語集と非公開用語集

OmegaT から TaaS の公開データを利用する分には、特に登録不要です。

非公開データを利用するには、まずウェブブラウザで https://demo.taas-project.eu/account/keys/create?system=omegaT にアクセスして、キーを作成する必要があります。

作成したキーは、起動オプション -Dtaas.user.key=xxxxx で使用します。OmegaT ランチャー用設定ファイル（OmegaT.l4J.ini、omegat.kaptn、OmegaT.sh）に雛形の項目が含まれています。

プライベートキーを指定せずにサービスを利用しようとすると、次のメッセージがログに出力されます：TaaS API キーが指定されていません。https://demo.taas-project.eu/account/keys/create?system=omegaT にアクセスし、専用の API キーを作成してください。それを Java の起動オプション -Dtaas.user.key=xxxxx に指定して OmegaT を起動してください。(TAAS_API_KEY_NOT_FOUND)

3. TaaS サービスへのアクセス

［設定］→［用語集］をクリックし、次のオプションを表示します：

TaaS 用語集をダウンロード：現在のプロジェクトの原文言語と訳文言語に応じた既存の用語集を閲覧し、ダウンロードできます。非公開の用語集は、太字で表示されます。用語集をダウンロードすると、現在の用語集フォルダー内に TBX 形式で保存されます。

TaaS 用語を参照：チェックを入れると、分節を移動するたびに TaaS データを検索します。すべての用語集（公開、非公開とも）は、使用中の原文言語と訳文言語に限定して検索されます。

データ量を制限するには、TaaS 用語分野を選択から特定分野を選んでください。ダイアログ上で、すべての分野から用語を取得するか、特定の一分野からだけ取得するかを選択できます。

第21章 機械翻訳

1. はじめに

翻訳メモリを使ってユーザーが行う翻訳（OmegaT　が、そうですが）に対して、機械翻訳（MT）はルールベースや統計ベースの自動翻訳ツールです。翻訳メモリは使いません。統計的学習手法は、原文とその対訳を大量に解析して、翻訳モデルを構築したものです。機械翻訳サービスは、研究評価段階では良好で堅実な結果を達成しています。

機械翻訳サービスを利用するには、メニューから［設定］→［機械翻訳］を選び、任意のサービスを有効にします。これらはすべて、ウェブサービスです。利用する際は、インターネットに接続していなければなりません。

2. Google Translate

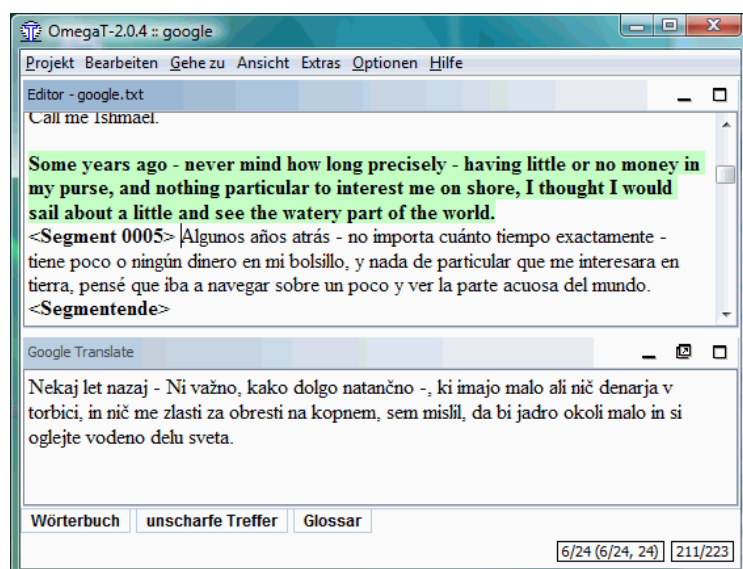
Google Translate は、Google が提供する有料サービスです。テキストやウェブサイト、ドキュメントを翻訳でき、その対応言語は増加し続けています。執筆時点で、アルバニア語からイディッシュまで　50 以上の言語に対応しており、もちろん主要な言語をすべて含んでいます。現在のバージョンは、利用に応じて、100 万字ごとに 20 US ドル課金されます。

重要：Google Translate API v2 は、サービスの利用を開始する前に、Google API コンソールで課金情報を設定する必要があります（詳しくは Pricing [https://developers.google.com/translate/v2/pricing?hl=en-US] を参照してください）。Google のサービスを利用するに当たっては、アカウント登録時に Google から送られてくるプライベート ユニークキーが必要になります。OmegaT 起動時にこのキーを設定する方法については、「OmegaT のインストールと実行」の「起動コマンドの引数」を参照してください。

Google Translate による翻訳の品質は、一方ではもちろん、訳文言語のテキストの蓄積量、また翻訳元となる言語がどれくらい適用できるかによります。しかし他方では、構築した翻訳モデルの品質にもよります。品質が十分でない場合もありますが、時が経てば必ず改善されるので、悪くはならないということが確実に言えます。

3. OmegaT ユーザーと Google Translate

機械翻訳の使用を OmegaT ユーザーに強要することは、ありません。使用する場合でも、その翻訳を受け入れるかの決定や、最終的な翻訳内容を Google に委ねるわけではありません。次のウィンドウには、a）英語の原文、b）スペイン語の訳文、c）スロヴェニア語の訳文、が表示されています。b と c は共に、Google Translate による翻訳です。

図21.1 Google Translate - 例

スペイン語の翻訳は、スロヴェニア語の翻訳よりも高品質です。動詞「interest」と「sail」はそれぞれ、スペイン語「interesar」と「navegar」に正しく翻訳されています。一方、スロヴェニア語の方は、どちらの単語も名詞として翻訳されています。この場合、スペイン語の翻訳は少なくとも部分的に、実際の書物の翻訳に基づいている可能性が高いと言えます。

いったん機械翻訳を有効にした後は、原文分節を新しく移動するたびに、機械翻訳ウィンドウにその翻訳の候補が表示されます。その候補を適用したい場合は、Ctrl+M を押すと、編集中の訳文分節が置換されます。上の例で言えば、Ctrl+M を押すと、スペイン語の訳文がスロヴェニア語の候補に置換されます。

Google 翻訳機能は、原文分節を Google へ送信します。これを望まない場合は、［設定］→［機械翻訳］→［Google Translate v2］のチェックを外してください。

いずれにしても、原文分節以外の情報は機械翻訳サービスに送信されません。オンライン版の Google 翻訳には、他の候補を表示したり、フィードバックを送信する機能があります。今のところ、これらの機能は OmegaT では使えません。

4. Belazar

Belazar [http://belazar.info/] は、ロシア語 - ベラルーシ語用の機械翻訳ツールです。

5. Apertium

Apertium [http://www.apertium.org/] はフリー / オープンソースの機械翻訳プラットフォームです。カタルーニャ語、スペイン語、アイルランド語、ポルトガル語、オック語、フランス語など、もともとは言語的に近い関係の言語を対象としていましたが、近年はもっと離れた言語間（たとえば英語 - カタルーニャ語）も扱えるように拡張されています。翻訳可能な最新の言語ペアの一覧は、ウェブサイトで確認してください。

プラットフォームが提供するもの

- 言語に依存しない機械翻訳エンジン
- 機械翻訳システム（特定の言語ペア向け）を構築するために必要な言語データの管理ツール
- 増え続ける言語ペア向けの言語データ

Apertium は浅い転移（shallow-transfer）の機械翻訳エンジンを使っています。これは入力テキストを、次のような工程の流れ作業に送り込みます。すなわち書式設定の解除、形態素解析、品詞の曖昧性除去、浅い構造化転移、語彙の転移、形態素生成、そして書式の再設定です。

Apertium を使うと、さまざまな言語ペア向けの機械翻訳システムを構築することができます。そのために必要な言語データは、単純な XML ベースの標準フォーマットで用意します（手入力でも、既存資産を変換したものでもよい）。このデータは提供されているツールによって、機械翻訳エンジンが高速に処理できる形式にコンパイルされます。

6. 機械翻訳 - トラブルシューティング

機械翻訳ウィンドウに何も表示されない場合は、以下の確認をしてください：

- ネットワークに接続していますか？機械翻訳を使用するためには、ネットワークに接続しておく必要があります。
- どの言語ペアを使おうとしていますか？有効にしたサービスが、その言語ペアに対応しているかどうか確認してください。
- Google Translate が動きません：Translate API サービス [https://developers.google.com/translate/v2/faq]に登録しましたか？Google Translate API サービスは無料ではない点に注意してください。詳しくは「OmegaT のインストールと実行」（起動引数）を参照してください。

- 「Google Translate returned HTTP response code: 403 ...」: pinfo.list ファイル（Mac の場合）に入力した 38 文字のキーが正しいか確認してください。また Translate API サービス [https://developers.google.com/translate/v2/faq]が有効になっているか、確認してください。

- 正しい Google API キーを入力したのに、Google Translate が動きません：［設定］→［機械翻訳］→［Google Translate v2］にチェックが入っているか確認してください。

- Google Translate v2 が「Bad request」しか表示しません：プロジェクトの原文言語と訳文言語の設定を確認してください。言語が未設定の場合に、このような結果が返ってきます。

第22章 綴り確認

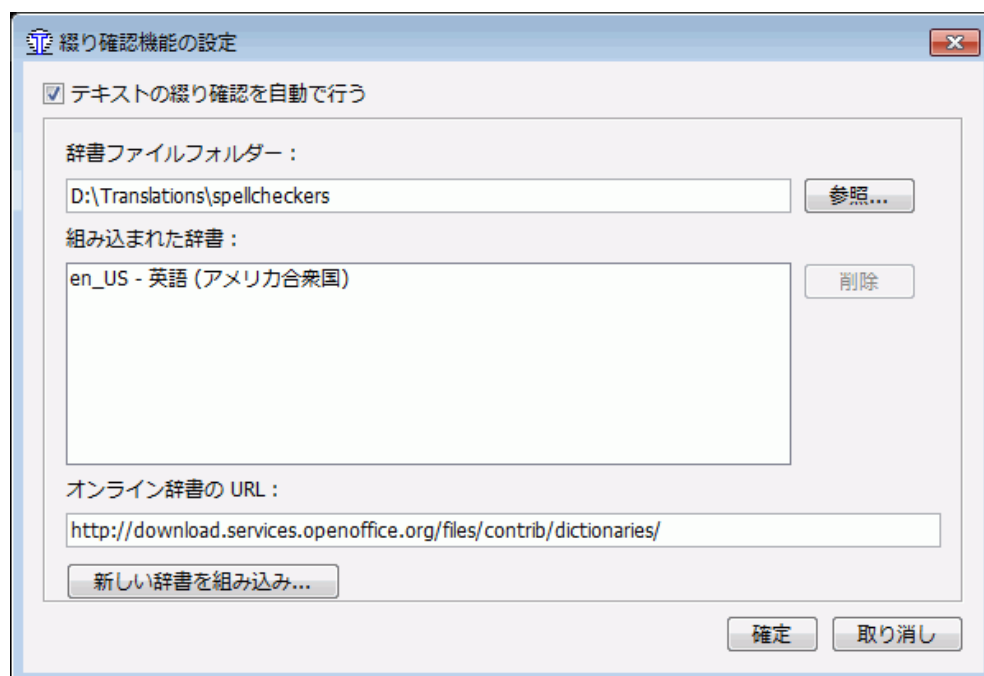
OmegaT は組み込みの綴り確認機能を備えています。これは Apache OpenOffice や LibreOffice、Firefox、Thunderbird で使われているものと同じ機能です。そのため、これらのプログラム向けに広く提供されているフリーの綴り確認用辞書を、OmegaT でも利用することができます。

1. 綴り確認用辞書の組み込み

綴り確認機能を使うためには、まず初めに適切な辞書（たとえば訳文言語用の）を組み込んでおく必要があります。辞書は、次の手順で組み込んでください：

- ファイルマネージャー上から、辞書を格納するための新しいフォルダーを適切な場所に作成します。以下の図では、「D:\Translations\spellcheckers」を指定しています。
- OmegaT で ［設定］→［綴り確認...］ を選び、辞書ファイルフォルダー欄の横にある［参照...］ボタンをクリックします。先ほど作成した辞書用フォルダーを指定します。
- 使いたい辞書ファイルを、そのフォルダーに配置します。これには基本的に 2 つの方法があります。ひとつは手動でファイルをコピーする方法です。コンピューター内の他の場所からファイルマネージャーを使ってコピーするとか、OmegaT の［新しい辞書を組み込み］ボタンを押して、利用可能な辞書の一覧から選ぶという形になります。後者の組み込みには、インターネットへの接続が必要になります。選んだ言語の辞書が組み込まれると、［綴り確認機能の設定］ウィンドウの［組み込まれた辞書］欄に表示されます（表示されるまでしばらく時間がかかります）。

手動でファイルをコピーする方法は、コンピューター内のどこかに使いたい辞書ファイルがある場合に適しています。たとえば Apache OpenOffice や LibreOffice、Firefox、Thunderbird をインストール済みの場合が該当します。もっとも、オンライン辞書の URL 欄を使って、インターネット上から辞書を探すほうが、簡単かもしれません。

図22.1 綴り確認機能の設定

［新しい辞書を組み込み］ボタンをクリックして［辞書の組み込み］ウィンドウを表示し、組み込みたい辞書を選んでください。

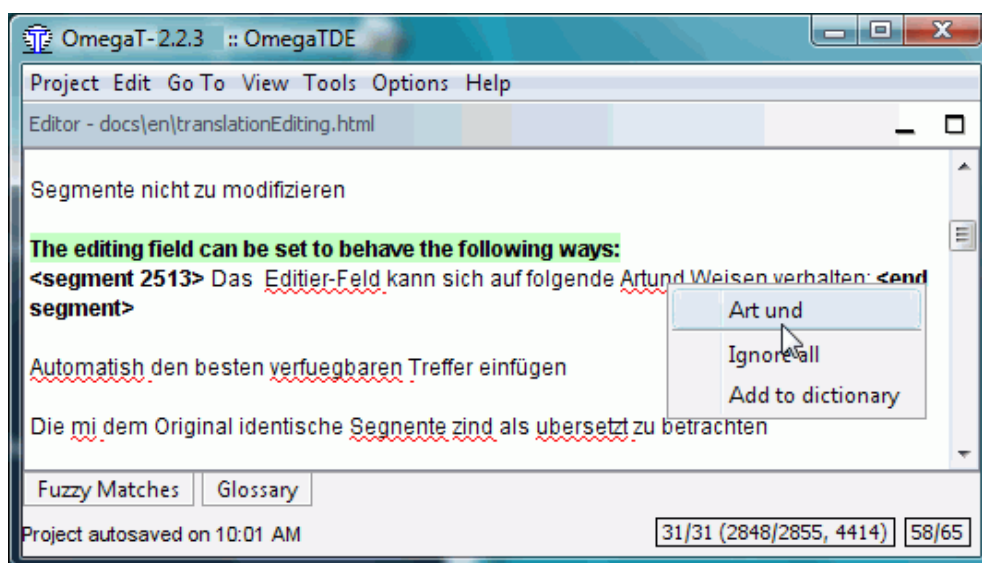
ファイル名は、プロジェクトの設定ダイアログ（［プロジェクト］→［プロジェクト設定...］）で設定されている訳文言語に対応した言語コードと同じ名前にしてください。たとえば、訳文言語が「ES-MX」（メキシコスペイン語）の場合、辞書ファイル名を「es_MX.dic」と「es_MX.aff」にする必要が

あります。もし通常のスペイン語辞書しか入手できなかった場合は、ファイル名を「es_es.dic」と「　es_es.aff」から、「es_MX.dic」と「es_MX.aff」に変更してください。そうすれば、綴り確認用辞書として使うことができます。この場合はもちろん、メキシコスペイン語ではなく、標準のスペイン語として綴りを確認することになる点に注意してください。

2. 綴り確認用辞書の使用

OmegaT に、どの辞書を使うか指示する必要はありません。OmegaT はプロジェクトの言語設定に基づいて、正しい言語用の辞書を使います。しかし、言語コードと同一の辞書ファイルが存在するかどうかは、確認してもいいでしょう。たとえばプロジェクトの訳文言語が「FR」に設定されている場合、「FR-FR」の辞書ファイルは使われません。必要であれば、辞書ファイルの名前を変更するか、または逆にプロジェクトの言語設定を変更してください。

綴り確認機能を有効にするには、［設定］→［綴り確認...］を選び、［テキストの綴り確認を自動で行う］チェックボックスにチェックを入れます（上図を参照してください）。

図22.2 綴り確認機能の使用

下線が引かれた単語（上図では「Artund」）の上で右クリックすると、訂正候補（上図では「Art und」）が表示されます。加えて、検出された単語を［すべて無視］（Ignore　all）させることもできますし、［辞書へ追加］（Add to dictionary）することもできます。

3. 使用時のヒント

綴り確認機能が動作しない場合は、まず［綴り確認機能の設定］ダイアログ（［設定］→［綴り確認...］）の［テキストの綴り確認を自動で行う］チェックボックスにチェックが入っているか確認してください。

また、プロジェクトの設定から、訳文の言語コードと辞書の言語が一致しているかを確認してください。綴り確認機能は、訳文の言語コードから使用する言語を判別します。たとえば、訳文言語がブラジルポルトガル語（PT_BR）である場合、辞書が存在するフォルダーには次の 2 つのファイル pt_br.aff と pt_br.dic が存在する必要があります。

すでにテキストの大部分を翻訳した後で、プロジェクト設定中の訳文の言語コードが辞書の言語と一致していないことに気づいた場合（たとえば言語コードが pt_BR であるのに、pt_BR 用の辞書が存在しない場合）は、単純に 2 つの辞書ファイルの名前を変更すれば問題ありません（pt_PT.aff と pt_PT.dic をそれぞれ pt_BR.aff と pt_BR.dic に変更）。もちろん、少し休憩を取ってから、適切な辞書ファイルをダウンロードしたほうがより賢明です。

［削除］ボタンを押すと、選択した辞書は物理的に削除されます。その辞書を他のプログラムと共有していた場合は、他のプログラムからも参照できなくなってしまいます。理由はさておき、ときどき削除をするのであれば、念のため別のフォルダーにファイルのコピーを取っておいたほうがよいでしょう。

第23章 その他の話題

1. OmegaT コンソールモード

注記

興味のある上級者のユーザーのみを対象としています。

コンソール（コマンドラインとも）モードは、スクリプト環境で OmegaT を翻訳ツールとして用いる場合に使用できます。コンソールモードで開始すると、どのコンソール上でも動作するように、GUI を表示しません。そして、指定されたプロジェクトを自動的に処理します。たとえば、あるプログラム用のプロジェクトで、GUI をたくさんの言語に地域化する必要があるとします。コンソールモードを使うと、そのプログラムをビルドする工程に、インターフェースを地域化する作業を組み込むことができます。

1.1. 必要な条件

このモードで OmegaT を起動するためには、有効な OmegaT プロジェクトが存在している必要があります。場所はどこであってもかまいません。起動時にコマンドライン上で指定します。

標準とは別の設定を指定したい場合は、有効な設定ファイル（filters.conf と segmentation.conf）も用意する必要があります。その場合は、次の 2 つのうちいずれかの方法をとってください：

- OmegaT を通常通り（GUI 経由で）起動して設定しておきます。そのままコンソールモードで起動すると、同じ設定を使用します。
- OmegaT を通常通り起動できない（GUI 環境がない）場合は、次の方法をとってください：同じ端末の他の場所、または別の端末にある OmegaT の設定ファイルを、あらかじめ決めたフォルダーにコピーしておきます。フォルダーの場所は、起動時にコマンドライン上で指定するので、どこであってもかまいません（次の項を参照してください）。設定ファイルは、filters.conf と segmentation.conf です。ユーザーのホームフォルダー（Windows の場合 C:\Documents and Settings\%User%\OmegaT、Linux の場合 %user%/.omegat/）にあります。

1.2. コンソールモードで起動する

コンソールモードで OmegaT を起動する場合、いくつかの引数が必要になります。最も重要な引数は、<project-dir> です。他に任意の引数として、--config-dir=<config-dir> もあります。例：

$> java -jar OmegaT.jar /path/to/project \

--config-dir=/path/to/config-files/ \

--mode=console-translate \

--source-pattern={正規表現} \

--tag-validation=[block|warn]

説明：

- <project-dir> プロジェクトの場所を指定します。この引数を指定すると、OmegaT がコンソールモードで起動し、指定されたプロジェクトを翻訳します。
- --config-dir=<config-dir> 設定ファイルの場所を指定します。このオプションが指定されない場合は、OmegaT は初期設定を使用します（ユーザーホームフォルダーの下の OmegaT フォルダーか、それが存在しない場合は現在の作業フォルダー）。
- --mode=console-translate この引数を指定すると、OmegaT がコンソールモードで起動し、指定されたプロジェクトを翻訳します。

- --source-pattern={正規表現} 翻訳したいファイルを指定します。正規表現を指定した例：test\.html
- --tag-validation=[abort|warn] 「abort」を指定した場合は、不正なタグを見つけた時点で、タグ検証を中止します。「warn」を指定した場合は、エラーを出力しますが、プログラムは処理を継続します。上記以外の値を指定した場合はすべて、タグ検証を行いません。

1.3. サイレントオプション

コンソールモード用の追加設定として --quiet があります。このオプションを指定すると、画面出力が極力抑えられます。通常ステータスバーに表示されるようなメッセージは、表示されなくなります。

使い方: java -jar OmegaT.jar /path/to/project --mode=console-translate --quiet

1.4. タグ検証の設定

もう 1 つ別の追加のコンソールモードのオプションとして、--tag-validation=[abort|warn] があります。このオプションを指定すると、翻訳や整列の処理時にタグ検証を行います。オプション値「abort」を指定した場合は、タグのエラーがあった場合はエラーを出力して処理を中止します。オプション値「warn」を指定した場合は、タグのエラーがあった場合はエラーを出力して処理を継続します。

使い方: java -jar OmegaT.jar /path/to/project --mode=console-translate --tag-validation=abort

2. Java properties ファイルの自動整列機能

OmegaT は Java の .properties ファイルをコンソールモードで整列することができます。すなわち、原文とその対訳の Properties ファイルから、対応する TMX ファイルを生成します。使い方：

java -jar OmegaT.jar --mode=console-align /my-project-dir --alignDir=/translatedFiles/

alignDir には、プロジェクトの訳文言語に翻訳された訳文ファイルを配置しておく必要があります。たとえばプロジェクトが英→日翻訳の言語設定になっている場合、alignDir フォルダーには、ファイル名が「_ja」で終わるバンドルが存在しなければなりません。整列した TMX ファイルは、align.tmx という名前で omegat フォルダーに出力されます。

3. 字体の設定

このダイアログでは、OmegaT の以下のウィンドウで使用される字体を変更できます：

- メインウィンドウ（編集ウィンドウ、参考訳文ウィンドウ、用語集ウィンドウ）
- 検索ウィンドウ
- タグ検証ウィンドウ

このダイアログはメインメニューの 設定 → 字体... を選択すると表示できます。このダイアログは以下の項目からなります：

- 字体名：実行環境で利用可能な字体一覧をドロップダウン表示するコンボボックスです。
- 大きさ：字体の大きさを変更します。
- サンプルテキスト：選択した字体をその場でプレビューできる領域です。

注意：状況によっては、字体設定の変更後、その反映に時間がかかることがあります。これは特に、多くの分節を含む巨大なファイルを編集ウィンドウで開いているときや、負荷の高い処理を並列実行してハードウェアが遅くなっているときに、見られます。また、特定の言語向けにデザインされた字体もあるため、字体によってはある言語との組み合わせで特に効果的に使用できる場合もあります。アルファベットや文字体系が大きく異なる言語間（たとえばロシア語と日本語のような）で翻訳をしている場合は、両方に適した字体を選ぶことが大切です。

4. データを失わないために

OmegaT は堅牢なプログラムです。しかし、他のプログラムを使用するときと同様、万が一データが消えてしまった場合の対策を考えておくべきでしょう。OmegaT は、翻訳作業のすべての内容を翻訳メモリ（omegat フォルダーの project_save.tmx ファイル）に保存します。

また、プロジェクトを開くか再読み込みするたびに、翻訳メモリのバックアップファイル（同じく omegat フォルダーの project_save.tmx.YYYYMMDDHHNN.bak ファイル）を作成します。「YYYYMMDDHHNN」の部分は、前回保存した日付を 4 桁の年（YYYY）、月（MM）、日（DD）、時（HH）、分（NN）で表します。

翻訳データが消えた場合、以下の手順にしたがって、直近に保存されたバックアップ（デフォルトの設定であれば過去 10 分以内）からプロジェクトを復元してください:

1. プロジェクトを閉じる
2. 現在の project_save.tmx ファイルの名前を一時的に変更する（たとえば project_save.tmx.temporary）
3. 探している内容ともっとも近い翻訳メモリのバックアップファイルを選択する
4. そのファイル名を project_save.tmx に変更する
5. プロジェクトを再度開く

大切なデータを失わないためには：

- omegat フォルダーにある project_save.tmx ファイルを、CD や DVD のような外部メディアや、クラウド ストレージに、定期的にコピーをとっておきましょう。
- OmegaT の使用に慣れるまでは、翻訳ファイルを定期的に作成し、翻訳後のファイルに最新の翻訳内容が反映されているかを、こまめに確認してください。
- 翻訳作業全体の途中で source にあるファイルを変更した場合は、特に注意してください。翻訳を開始した後に原文ファイルを変更すると、OmegaT は既に翻訳した分節を見つけられなくなるかもしれません。
- まず、このヘルプに書かれた内容を参照して、解決を試みてください。それでも問題が解決しない場合は、OmegaT ユーザーグループ [http://tech.groups.yahoo.com/group/OmegaT/]に投稿してください。多言語に対応しているので、自分が一番書きやすい言語（日本語でも）で投稿してかまいません。

付録A 言語設定 - ISO 639 コード一覧

言語コードに関する詳細な情報は ISO 639 コード表 [http://www.sil.org/ISO639-3/codes.asp] を参照してください。（日本語訳注：言語の日本語表記は http://ja.wikipedia.org/wiki/ISO_639 を参照しました。）

表A.1 ISO 639-1/639-2 言語コード一覧

言語名	ISO 639-1	ISO 639-2
アブハズ語（Abkhaz）	ab	abk
アファル語（Afar）	aa	aar
アフリカーンス語（Afrikaans）	af	afr
アカン語（Akan）	ak	aka
アルバニア語（Albanian）	sq	sqi
アムハラ語（Amharic）	am	amh
アラビア語（Arabic）	ar	ara
アラゴン語（Aragonese）	an	arg
アルメニア語（Armenian）	hy	hye
アッサム語（Assamese）	as	asm
アヴァール語（Avaric）	av	ava
アヴェスター語（Avestan）	ae	ave
アイマラ語（Aymara）	ay	aym
アゼルバイジャン語（Azerbaijani）	az	aze
バンバラ語（Bambara）	bm	bam
バシキール語（Bashkir）	ba	bak
バスク語（Basque）	eu	eus
ベラルーシ語（Belarusian）	be	bel
ベンガル語（Bengali）	bn	ben
ビハール語（Bihari）	bh	bih
ビスラマ語（Bislama）	bi	bis
ボスニア語（Bosnian）	bs	bos
ブルトン語（Breton）	br	bre
ブルガリア語（Bulgarian）	bg	bul
ビルマ語（Burmese）	my	mya
カタルーニャ語（Catalan）	ca	cat
チャモロ語（Chamorro）	ch	cha
チェチェン語（Chechen）	ce	che
チチェワ語、チェワ語、ニャンジャ語（Chichewa, Chewa, Nyanja）	ny	nya
中国語（Chinese）	zh	zho
チュヴァシ語（Chuvash）	cv	chv
コーンウォル語（Cornish）	kw	cor
コルシカ語（Corsican）	co	cos
クリー語（Cree）	cr	cre
クロアチア語（Croatian）	hr	hrv

言語名	ISO 639-1	ISO 639-2
チェコ語（Czech）	cs	ces
デンマーク語（Danish）	da	dan
ディベヒ語、モルディブ語（Divehi, Dhivehi, Maldivian）	dv	div
オランダ語（Dutch）	nl	nld
ゾンカ語（Dzongkha）	dz	dzo
英語（English）	ja	eng
エスペラント語（Esperanto）	eo	epo
エストニア語（Estonian）	et	est
エウェ語（Ewe）	ee	ewe
フェロー語（Faroese）	fo	fao
フィジー語（Fijian）	fj	fij
フィンランド語（Finnish）	fi	fin
フランス語（French）	fr	fra
フラ語、プラール語（Fula, Fulah, Pulaar, Pular）	ff	ful
ガリシア語（Galician）	gl	glg
グルジア語（Georgian）	ka	kat
ドイツ語（German）	de	deu
ギリシャ語（近代）（Greek, Modern）	el	ell
Guaraní	gn	grn
グジャラート語（Gujarati）	gu	guj
ハイチ語、ハイチクレオール語（Haitian, Haitian Creole）	ht	hat
ハウサ語（Hausa）	ha	hau
ヘブライ語（近代）（Hebrew (modern)）	he	heb
ヘレロ語（Herero）	hz	her
ヒンディー語（Hindi）	hi	hin
ヒリモトゥ語（Hiri Motu）	ho	hmo
ハンガリー語（Hungarian）	hu	hun
インターリングア（Interlingua）	ia	ina
インドネシア語（Indonesian）	id	ind
インターリング（Interlingue）	ie	ile
アイルランド語（Irish）	ga	gle
イボ語（Igbo）	ig	ibo
イヌピア語（Inupiaq）	ik	ipk
ido	io	ido
アイスランド語（Icelandic）	is	isl
イタリア語（Italian）	it	ita
イヌイット語（Inuktitut）	iu	iku
日本語（Japanese）	ja	jpn
ジャワ語（Javanese）	jv	jav

言語名	ISO 639-1	ISO 639-2
カラーリット語、グリーンランド語（Kalaallisut, Greenlandic）	kl	kal
カンナダ語（Kannada）	kn	kan
カヌリ語（Kanuri）	kr	kau
カシミール語（Kashmiri）	ks	kas
カザフ語（Kazakh）	kk	kaz
クメール語（Khmer）	km	khm
キクユ語（Kikuyu, Gikuyu）	ki	kik
ルワンダ語（Kinyarwanda）	rw	kin
キルギス語（Kirghiz, Kyrgyz）	ky	kir
コミ語（Komi）	kv	kom
コンゴ語（Kongo）	kg	kon
朝鮮語（Korean）	ko	kor
クルド語（Kurdish）	ku	kur
クワニャマ語（Kwanyama, Kuanyama）	kj	kua
ラテン語（Latin）	la	lat
ルクセンブルク語（Luxembourgish, Letzeburgesch）	lb	ltz
ガンダ語（Luganda）	lg	lug
リンブルフ語（Limburgish, Limburgan, Limburger）	li	lim
リンガラ語（Lingala）	ln	lin
ラーオ語（Lao）	lo	lao
リトアニア語（Lithuanian）	lt	lit
ルバ· カタンガ語（Luba-Katanga）	lu	lub
ラトビア語（Latvian）	lv	lav
マン島語（Manx）	gv	glv
マケドニア語（Macedonian）	mk	mkd
マダガスカル語（Malagasy）	mg	mlg
マレー語（Malay）	ms	msa
マラヤーラム語（Malayalam）	ml	mal
マルタ語（Maltese）	mt	mlt
マオリ語（Māori）	mi	mri
マラーティー語（Marathi (Marāṭhī)）	mr	mar
マーシャル語（Marshallese）	mh	mah
モンゴル語（Mongolian）	mn	mon
ナウル語（Nauru）	na	nau
ナヴァホ語（Navajo, Navaho）	nv	nav
ブークモール（ノルウェー語）（Norwegian Bokmål）	nb	nob
北ンデベレ語（North Ndebele）	nd	nde

言語名	ISO 639-1	ISO 639-2
ネパール語（Nepali）	ne	nep
ンドンガ語（Ndonga）	ng	ndo
ニーノシュク（ノルウェー語）（Norwegian Nynorsk）	nn	nno
ノルウェー語（Norwegian）	no	nor
ノス語（彝語）（Nuosu）	ii	iii
南ンデベレ語（South Ndebele）	nr	nbl
オック語（Occitan）	oc	oci
オジブワ語（Ojibwe, Ojibwa）	oj	oji
古代教会スラヴ語（Old Church Slavonic, Church Slavic, Church Slavonic, Old Bulgarian, Old Slavonic）	cu	chu
オロモ語（Oromo）	om	orm
オリヤー語（Oriya）	or	ori
オセット語（Ossetian, Ossetic）	os	oss
パンジャーブ語（Punjabi）	pa	pan
パーリ語（Pāli）	pi	pli
ペルシア語（Persian）	fa	fas
ポーランド語（Polish）	pl	pol
パシュトー語（Pashto, Pushto）	ps	pus
ポルトガル語（Portuguese）	pt	por
ケチュア語（Quechua）	qu	que
レト・ロマン語（Romansh）	rm	roh
ルンディ語（Kirundi）	rn	run
ルーマニア語（Romanian, Moldavian, Moldovan）	ro	ron
ロシア語（Russian）	ru	rus
サンスクリット（Sanskrit (Saṁskṛta)）	sa	san
サルデーニャ語（Sardinian）	sc	srd
シンディー語（Sindhi）	sd	snd
北部サーミ語（Northern Sami）	se	sme
サモア語（Samoan）	sm	smo
サンゴ語（Sango）	sg	sag
セルビア語（Serbian）	sr	srp
スコットランド・ゲール語（Scottish Gaelic, Gaelic）	gd	gla
ショナ語（Shona）	sn	sna
シンハラ語（Sinhala, Sinhalese）	si	sin
スロバキア語（Slovak）	sk	slk
スロベニア語（Slovene）	sl	slv
ソマリ語（Somali）	so	som
南ソト語（Southern Sotho）	st	sot

言語名	ISO 639-1	ISO 639-2
スペイン語、カスティーリャ語（Spanish, Castilian）	es	spa
スンダ語（Sundanese）	su	sun
スワヒリ語（Swahili）	sw	swa
スワジ語（Swati）	ss	ssw
スウェーデン語（Swedish）	sv	swe
タミル語（Tamil）	ta	tam
テルグ語（Telugu）	te	tel
タジク語（Tajik）	tg	tgk
タイ語（Thai）	th	tha
ティグリニャ語（Tigrinya）	ti	tir
チベット語、中央チベット語（Tibetan Standard, Tibetan, Central）	bo	bod
トルクメン語（Turkmen）	tk	tuk
タガログ語（Tagalog）	tl	tgl
ツワナ語（Tswana）	tn	tsn
トンガ語（トンガ）（Tonga (Tonga Islands)）	to	ton
トルコ語（Turkish）	tr	tur
ツォンガ語（Tsonga）	ts	tso
タタール語（Tatar）	tt	tat
トウィ語（Twi）	tw	twi
タヒチ語（Tahitian）	ty	tah
ウイグル語（Uighur, Uyghur）	ug	uig
ウクライナ語（Ukrainian）	uk	ukr
ウルドゥー語（Urdu）	ur	urd
ウズベク語（Uzbek）	uz	uzb
ヴェンダ語（Venda）	ve	ven
ベトナム語（Vietnamese）	vi	vie
ヴォラピュク（Volapük）	vo	vol
ワロン語（Walloon）	wa	wln
ウェールズ語（Welsh）	cy	cym
ウォロフ語（Wolof）	wo	wol
フリジア語（Western Frisian）	fy	fry
コサ語（Xhosa）	xh	xho
イディッシュ語（Yiddish）	yi	yid
ヨルバ語（Yoruba）	yo	yor
チワン語（Zhuang, Chuang）	za	zha
ズールー語（Zulu）	zu	zul

付録B 編集ウィンドウのキーボードショートカット

ここでは、編集ウィンドウ上で使用できる、各キーに割り当てられた動作について説明します。以下で「分節内に移動します」という説明が出てきた場合は、カーソルが分節よりも前にあれば、分節の先頭に移動することを、カーソルが分節の後にあれば、分節の末尾に移動することを指しています。

表B.1 編集ウィンドウのキー動作

キーの組み合わせ	動作
Left	一文字分、左に移動します。ただし分節の先頭よりも左には移動しません。
Right	一文字分、右に移動します。ただし分節の末尾よりも右には移動しません。
Ctrl+Left	一単語分、左に移動します。ただし分節の先頭よりも左には移動しません。
Ctrl+Right	一単語分、右に移動します。ただし分節の末尾よりも右には移動しません。
PgUp	文書を一画面分、上方向にスクロールします。
PgDn	文書を一画面分、下方向にスクロールします。
Home*	分節内で、行頭に移動します。
End*	分節内で、行末に移動します。
Ctrl+Home	分節の先頭に移動します。
Ctrl+End	分節の末尾に移動します。
Ctrl+PgUp	文書の先頭に移動します（Mac：Cmd+PgUp）。
Ctrl+PgDn	文書の末尾に移動します（Mac：Cmd+PgDn）。
Backspace*	カーソルの直前の文字を削除します。
Delete*	カーソルの直後の文字を削除します。
Ctrl+Backspace	カーソルの直前から現在の単語の先頭の文字までを削除します（Mac：Alt+Backspace）。
Ctrl+Delete	カーソルの直後から次の単語の直前の文字までを削除します（Mac：Alt+Delete）。
Ctrl+Enter	前の分節へ移動します（Mac：Cmd+Enter）。
Ctrl+A	分節全体を選択します（Mac：Cmd+A）。
Ctrl+Shift+O	文字列の方向（右から左、左から右）の切り替え
Ctrl+Space	入力補完用のダイアログボックスを表示します（Mac：Esc）。Ctrl+PgUp/PgDn（Mac では Cmd+PgUp/PgDn）を押すことで、用語集、定型文、未使用のタグ、文字一覧を連続的に切り替えることができます。

* これらのキーでは、カーソルが編集中の分節の外側にあると、動作が異なります：

- Home：編集中の分節の先頭に移動します。
- End：編集中の分節の末尾に移動します。
- Backspace：何もしません。
- Delete：何もしません。

- 任意の文字キー：カーソルが編集中の分節の外側にある場合は、無視されます。

［ Shift ］キー自体は、特に機能を持ちません。［ Shift ］キーを押しながら何かのキーを押しても、そのキー本来の動作をするだけです。ただし Shift+Enter の組み合わせに限っては、テキストに改行を挿入します。

システム共通のコマンド：すべて選択（ Ctrl+A ）、貼り付け（ Ctrl+V ）、切り取り（ Ctrl+X ）、コピー（ Ctrl+C ）、参考訳文を挿入（ Ctrl+I ）、原文を挿入（ Ctrl+Shift+I ）は、原理的に現在編集中の分節内のテキストに対してだけ実行されます。

現在のウィンドウから別のウィンドウに移動する（たとえば編集ウィンドウから参考訳文ウィンドウに）には、Ctrl+Tab を使います。Ctrl+Shift+Tab を押すと、ひとつ前のウィンドウへ戻ります。ショートカット Ctrl+A と Ctrl+C の連携は、ウィンドウ内のすべてのテキストまたは一部のテキストをクリップボードにコピーします。

ショートカットは個人設定でカスタマイズできることに注意してください。詳しくは付録 ショートカットのカスタマイズを参照してください。

付録C OmegaT チームプロジェクト

1. バージョン管理 - はじめに

OmegaT はバージョン管理に基づいた共同翻訳機能を提供します。バージョン管理はソフトウェアの分野で広く使われている手法で、プログラムのソースコードに関する変更を維持･管理し、チームによる共同開発作業を強力に支援します。OmegaT は非常に普及している 2 つのバージョン管理システム（以下、VCS）に対応しています。Apache Subversion [http://subversion.apache.org/]（コマンド名が svn であることから、しばしば SVN の略称で呼ばれる）と、Git [http://git-scm.com/] です。翻訳者のチームが VCS を使うことには、次のような利点があります：

- 複数のチームメンバーが、お互いに邪魔することなく、同じ翻訳プロジェクトで同時に作業できます
- プロジェクトの翻訳メモリや用語集などを、共有できます
- デフォルトでは 3 分ごとに、最新データがチームメンバーに共有されます。
- システムが、共有データのバージョン管理を行います
- 競合（たとえば、同じ分節や用語を複数人が同時に編集している状態）を監視して、マージし、解決できます

以下は、本章で使用される用語の簡単な説明です：

- VCS サーバー：ここでは、SVN や Git のサーバーを指します。ネットワーク上で共有データを維持･管理する環境です。サーバーはローカルネットワーク上にも存在できますが、たいていの場合はインターネット上に存在し、URL を指定してアクセスします。少なくともチームの誰か一人は、プロジェクトの管理者として、サーバー側の管理に通じている必要があります。環境設定や OmegaT プロジェクトのインポート、メンバーに対するアクセス権の賦与、競合の解決などが仕事です。
- VCS クライアント：サーバーへのインターフェースです。プロジェクトの管理者は、SVN または Git 用のクライアントをコンピューターにインストールして、OmegaT プロジェクトを管理します。Windows 向けの汎用クライアントとしては、TortoiseSVN [http://tortoisesvn.net/] や TortoiseGit [http://code.google.com/p/tortoisegit/] が有名です。Linux や Mac OS X など、他の OS 向けにも同様のパッケージが存在します。
- リポジトリ：データを維持･管理する場所の単位です。ローカルネットワークに存在する場合もあれば、インターネット上の場合もあります。プロジェクトメンバーが、VCS クライアントを介して接続する先が、これになります。
- チェックアウト：ローカルコンピューター上にリポジトリの作業コピーを作成する操作を指します。（日本語訳註：Git では同様の操作を「クローン」と呼びます）サーバーは、チェックアウト情報を保持します。これによって、後でコミット（下記を参照）がうまく機能します。
- コミット：チェックアウトしたデータをローカルで編集して、あるていどの区切りが付いたら、リポジトリにコミットします。これによって、チームの他のメンバーにも最新のデータを共有可能になります。すべての衝突や競合（複数のメンバーがチェックアウトした同じデータを編集している状態）が解決されるまで、サーバーはコミットを受け付けません。
- 管理者：リポジトリの作成と維持の責任者。サーバー側の作業を担当します。問題を避けるために、最初のうちは決まったひとりの人が担当するとよいでしょう。
- ユーザー：プロジェクトで共同翻訳するチームのメンバー。

2. SVN でプロジェクトを共有する

SVN サーバーを使うには、2 つの選択肢があります。自前のサーバーに SVN をインストールするか、ホスティングサービスを利用するか、です。外部サービスを利用するなら、元の文書を外部のサーバー

にアップロードする前に、機密保持の観点から問題が無いか、よく検討した方が良いでしょう。この問題を避けるために、プライベートな SVN サーバーを立てるという手もあります。たとえば、すでに Apache サーバーを管理しているのであれば、VisualSVN のようなプログラムをインストールすれば済むかもしれません。

SVN サーバーを利用可能になったら、プロジェクト用のファイルを管理できます。管理者はそのために、SVN クライアントをローカルコンピューター上にインストールしておく必要があります。Windows 向けには、TortoiseSVN [http://tortoisesvn.net/] を勧めます。Mac 向けのクライアントは SourceForge [https://sourceforge.net/projects/macsvn/] などで見つけることができます。Linux 向けの情報は、『Subversion Commands and Scripts [www.yolinux.com/TUTORIALS/Subversion.html]』（英語）を参照してください。

2.1. リポジトリの作成

ここでの説明には、ProjectLocker [http://projectlocker.com/]（フリープランは 2 ユーザーまで）の SVN サーバーを例に取ることにします。（日本語訳注：日本語で使える同様のサービスには、フリープランが 10 ユーザーまでの Backlog（www.backlog.jp）などがあります）リポジトリの作成者は、暗黙的に管理者権限を持つことになりますので、注意してください。まずサイトにログインします。サイトを利用するのがはじめての場合は、ユーザー登録をしましょう。後で必要になるので、ユーザー名とパスワードは忘れないようにしてください。

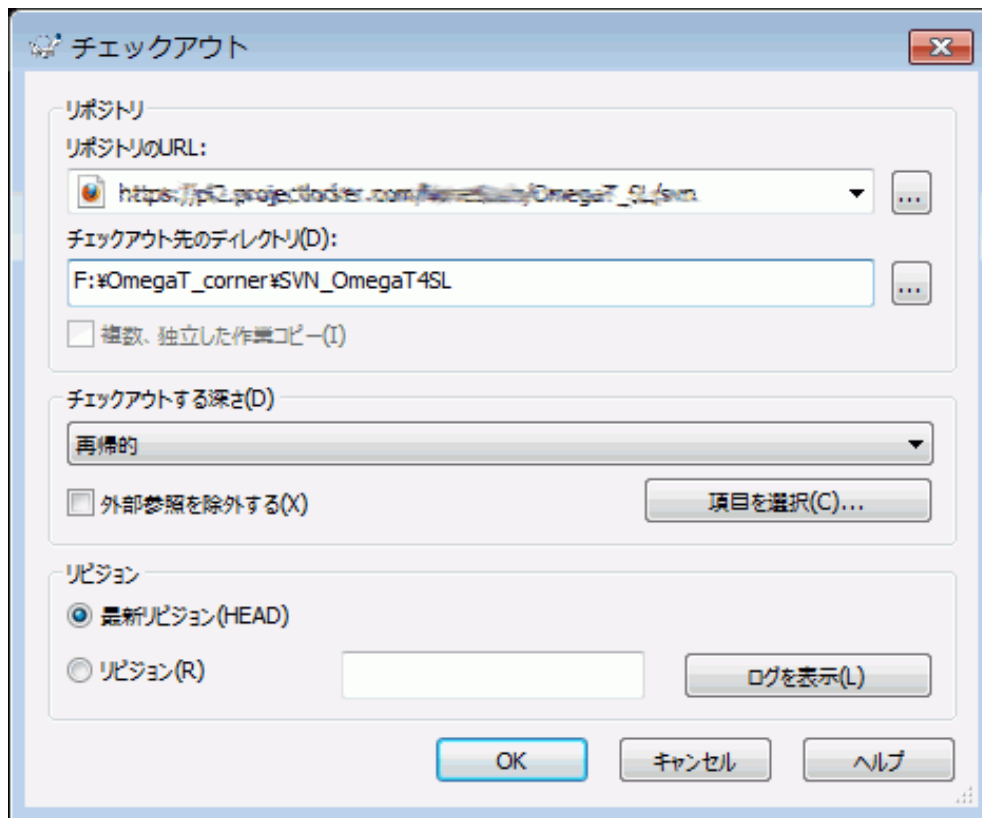
1. ProjectLocker で新しいプロジェクトを作成します
2. リポジトリの名前と説明を入力します（この例では「OmegaT」と「OmegaT SL Localization」と入力したものとします）
3. 「SVN」を選びます
4. 「Create Project」をクリックします

自分のアカウントの Projects ビューを開きましょう。プロジェクトサービスの下に表示されている URL は、SVN クライアントからサーバーに接続する際に使います。同じ画面から、チームメンバーを追加したり、権限を割り当てることができます。チームメンバーを追加するには、そのメンバーが先にユーザー登録を済ませている必要があります。ProjectLocker のフリープランは、1 プロジェクト 2 ユーザーまでなので、注意してください。

プロジェクトは、自分のニーズと開発スタイルに合わせて管理できます。OmegaT プロジェクトの場合と同様に、言語ペアごとにリポジトリを分けてもかまいません。言語ペアが同じであっても、依頼主や案件が違うなら、やはりリポジトリを分けた方がよいでしょう。あるいは、ひとつのリポジトリ内で「Project1」、「Project2」のようにサブフォルダーを作って分ける方法もあります（日本語訳注：この方法は、SVN でのみ有効です。Git では 1 リポジトリを部分使用できないため、たとえば前述の Backlog ではフリープランでも Git のリポジトリ数は制限されません）。そして共有したいデータを tm フォルダー、glossary フォルダー、dictionary フォルダーに置きます。

以下では、説明を単純にするために、ひとつの OmegaT プロジェクトに対してひとつのリポジトリを使うことにします。

2.2. SVN リポジトリにプロジェクトをインポートする - Windows の場合

先ほど作成したリポジトリは、まだ空の状態です。では、まずローカルコンピューターのディスク上に空のフォルダーを作成しましょう。プロジェクトを配置するための空のフォルダーを作成したら、その上で右クリックします。TortoiseSVN をインストールしてあれば、コンテキストメニューに [SVN チェックアウト(K)...] というコマンドが出てきます。それを選ぶと、下図のようなダイアログが表示されます。

ProjectLocker から取得した URL を [リポジトリのURL] 欄に入力します。 [チェックアウト先のディレクトリ] 欄が正しい (先ほど作成した空のフォルダーが指定されている) ことを確認し、 [OK] ボタンを押します。この操作が完了すると、フォルダーは作業コピーとなって、サーバーの最新の状態に同期されます。フォルダーのアイコンの上には、緑色の OK バッジが重なって表示されます。またフォルダーの中には、管理用の .svn フォルダーが作成されています (ただし隠しフォルダーなので、シェルの設定によっては表示されません) 。

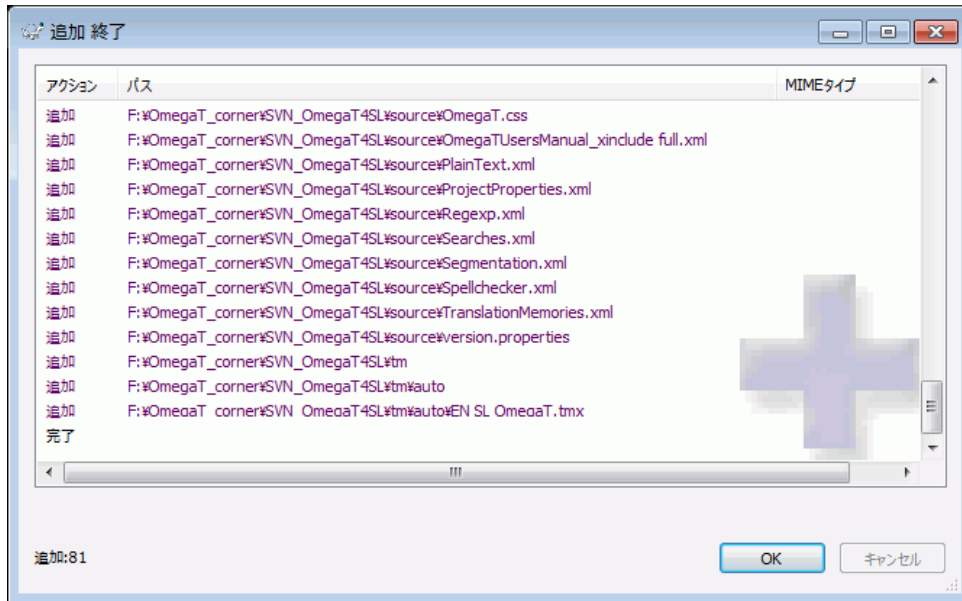
次の手順として、作業コピーフォルダーに OmegaT のファイルを追加しましょう。以下は、チームメンバーと共有するファイルです。チームプロジェクトは、これらのファイルを含みます :

- omegat プロジェクトファイル -- omegat.project
- 翻訳メモリファイル -- omegat\project_save.tmx
- source フォルダー内の原文ファイル
- プロジェクト専用のフィルター定義ファイル (省略可) -- omegat\filters.xml

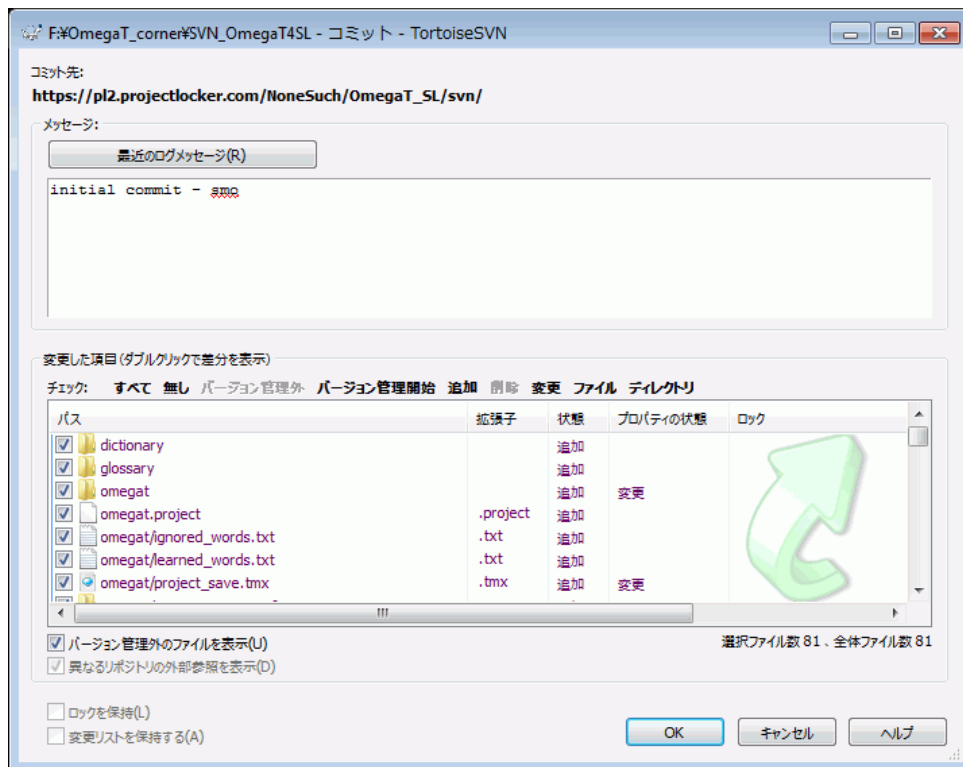
管理者は、次のフォルダーとその内容についても、同様に含めることができます : tm フォルダー、glossary フォルダー、dictionary フォルダー。また omegat フォルダーの中の ignored_words.txt と learned_words.txt も、チームで共有して維持していくに値します。omegat フォルダー内の .bak ファイルと project_stats.txt および project_stats_match.txt は、いかなる場合においても含めるべきではありません。これらのファイルを含めても、リポジトリが無意味に肥大化するだけで、何の役にも立ちません。target フォルダーとその内容についても、同じ方針でよいでしょう。

作業コピーフォルダーに必要なファイルをコピーしたら、フォルダーのアイコンが変化することに気づくでしょう。緑の OK バッジだったものが、作業コピーが変更されたことを示す赤い感嘆符に変わったはずです。次の 2 つの手順で、作業コピーの変更をサーバーに反映させます :

- 作業コピーフォルダーにコピーしたファイルを「追加」します : 作業コピーフォルダーにファイルやフォルダーをコピーしただけでは、SVN の管理下に入りません。そのファイル/フォルダーを SVN が管理するよう、明示的に指定する必要があります。この操作を「追加」と呼びます。「追加」するには、ローカルにチェックアウトしたフォルダーを右クリックして、 [TortoiseSVN] → [追加(A)...] を選びます。ダイアログが開くので、デフォルトの設定のまま [OK] ボタンをクリックします。

（日本語訳註：右ドラッグでコピーすれば、一回の操作で追加まで完了します）「追加」が完了すると、下図のようなウィンドウが開きます。

チェックアウトしたフォルダーの中身は、次のような状態になります。

名前	更新日時	種類
.svn	2013/03/09 21:39	ファイル フォルダー
dictionary	2013/01/10 16:11	ファイル フォルダー
glossary	2013/02/26 12:26	ファイル フォルダー
omegat	2013/03/09 13:42	ファイル フォルダー
source	2013/02/26 19:14	ファイル フォルダー
tm	2013/01/10 16:10	ファイル フォルダー
omegat.project	2013/02/19 16:28	PROJECT ファイル

- ローカルの変更をサーバーにコミットします：作業コピーフォルダーを右クリックして、[SVN コミット(C)...] を選びます。下図のようなウィンドウが開きます。同期したい変更にチェックを付けます。この場合は、追加したファイルとフォルダーが対象です。

適切なログメッセージを入力して、［OK］ボタンを押します。コミットウィンドウが開き、処理の経過が表示されます。これはサーバー側のリポジトリに対する、はじめてのコミットになります。

- ローカルの作業コピーを「更新」します：「更新」とは、作業コピーに対してリポジトリの最新状態を反映させる操作を指します。作業コピーフォルダーの上で右クリックして、［SVN　更新(U)...］を選びます。作業コピーフォルダーを開き、リポジトリの最新の状態が反映されたことを確認してください。

名前	更新日時	種類
.svn	2013/03/09 21:39	ファイル フォルダー
dictionary	2013/01/10 16:11	ファイル フォルダー
glossary	2013/02/26 12:26	ファイル フォルダー
omegat	2013/03/09 13:42	ファイル フォルダー
source	2013/02/26 19:14	ファイル フォルダー
tm	2013/01/10 16:10	ファイル フォルダー
omegat.project	2013/02/19 16:28	PROJECT ファイル

3. OmegaT でチームプロジェクトを使う

管理者がチームプロジェクトの設定を済ませたら、メンバーに必要なのは OmegaT でそのチームプロジェクトをダウンロードすることだけです。初回は、メニューから［プロジェクト］→［チームプロジェクトをダウンロード...］を選びます。この操作によって、実際にはプロジェクトがローカルフォルダーにチェックアウトされます。認証情報は記憶されます。毎回入力する必要は、ありません。Linux の場合、もし OmegaT が認証情報の入力を毎回促してくるようであれば、［パスワードを暗号化しないで保存する］チェックボックスにチェックを入れてみてください。

２回目以降に使用する場合は、他の OmegaT プロジェクトとまったく同じように、チームプロジェクトを開くだけです。OmegaT はそれがチームプロジェクトであることを識別して、デフォルトでは 3 分ごとにすべて自動で同期します。

付録D トークナイザー

1. はじめに

トークナイザー（またはステマー）は、原文および翻訳メモリデータ内の単語の語尾変化を認識することによって、参考訳文の照合精度を改善します。また用語集の照合精度も改善します。

英語用のステマー（訳注：単語から接尾辞などを取り除く処理をステミングと呼ぶことから）を例にとってみましょう。「cats」という文字列（「catlike」や「catty」でもよいですが）は、「cat」を語幹としています。同様に「stemmer」や「stemming」、「stemmed」は「stem」を語幹としています。ステミングのアルゴリズムは、「fishing」や「fished」、「fish」、「fisher」などの単語を、語幹の「fish」という単語に収斂します。この処理は、語尾や語頭の変化形を持つ言語を扱う場合に、とても役立ちます。スロベニア語から例を拝借すると、「良い」という意味の単語は、文法的にまったく正しく次の変化形を取ります：

- lep, lepa, lepo - 単数形、男性形、女性形、中性形
- lepši, lepša, lepše . - 比較級、主格、男性形、女性形、中性形、それぞれに形容詞の複数形あり
- najlepših - 最上級、複数形、M,F,N（男性，女性，中性）の所有格

2. 言語設定

OmegaT のトークナイザーは、デフォルトで有効になっています。プロジェクトの原文と訳文の言語に応じて、OmegaT がトークナイザーを自動的に設定します。プロジェクト設定ウィンドウから、トークナイザーの種類（言語のトークナイザー）やバージョン（互換設定）を変更できます。

現在の言語に対して使用可能なトークナイザーが見つからない場合、OmegaT は代わりに Hunspell を使用します（このような場合は、該当する言語の Hunspell 辞書が導入済みであることを確認してください）。

互換性に関する注意点

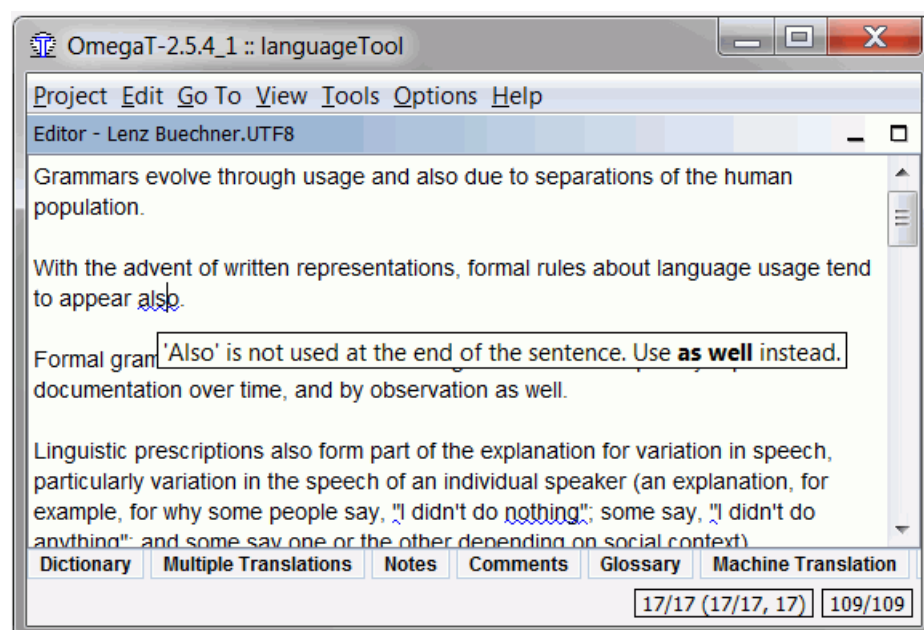
トークナイザーは、plugin フォルダーに古いバージョンのファイルが存在すると、正常に機能しません。plugin フォルダーにトークナイザーがあれば、OmegaT を起動する前にすべて削除してください。

付録E LanguageTool プラグイン

1. はじめに

LanguageTool [http://www.languagetool.org/] はオープンソースのスタイルと文法チェックツールです。英語、フランス語、ドイツ語、ポーランド語、オランダ語、ルーマニア語など数多くの言語に対応しています。詳しくは対応する言語の一覧 [http://www.languagetool.org/languages/]を参照してください。

LanguageTool は、単純な綴り確認機能では検出できないような間違いを検出するためのプログラムです。たとえば、there と their、no と now の取り違えなどが該当します。また、ある種の文法上の間違いも検出できます。綴り確認機能は含みません。LanguageTool は、言語固有の設定ファイルであらかじめ定義された規則に基づいて、間違いを見つけます。

図E.1 LanguageTool を OmegaT に組み込む

2. インストールと使い方

LanguageTool プラグインは OmegaT 本体に統合されました。［設定］→［言語チェッカー］にチェックを入れておくと、OmegaT は LanguageTool を自動的に利用します。プロジェクトの原文と訳文の言語に応じて、規則が（もしあれば）適用されます。規則が適用されると、編集ウィンドウ上の対応する語句には青い下線が引かれます（上の画像の「also」と「I didn't do nothing」に注目してください）。下線が引かれた語句の上にマウスカーソルをかざすと、説明が表示されます。

互換性に関する注意点

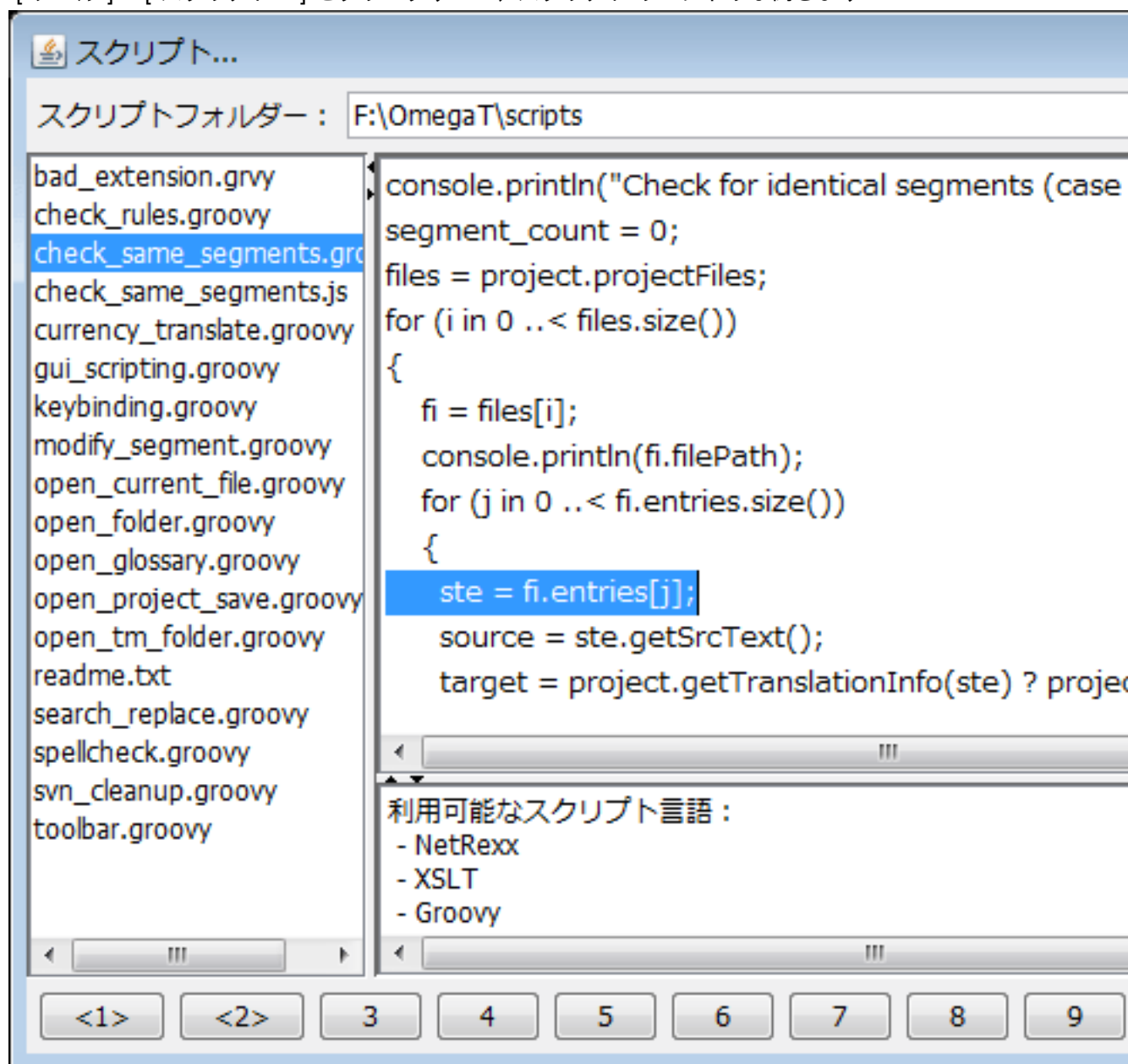
LanguageTool は、plugin フォルダーに古いバージョンのファイルが存在すると、正常に機能しません。plugin フォルダーに LanguageTool があれば、OmegaT を起動する前に削除してください。

付録F スクリプト

1. はじめに

さまざまなスクリプト言語で書かれたスクリプトを OmegaT で実行することができます。

2. 使い方

[ツール] → [スクリプト...] をクリックすると、スクリプト ウィンドウが開きます :

スクリプト　ウィンドウ上では、既存のスクリプトをテキスト欄に読み込んだり、いま開いているプロジェクトに対して実行できます。スクリプトをカスタマイズする手順は、次の通りです :

- 左側のスクリプト一覧から目的のスクリプト名をクリックして、編集ウィンドウに読み込みます。
- 下部にある「1」から「12」のどれかのボタンの上で右クリックし、 [スクリプトを設定] を選びます。上の画像では、 2 つのスクリプトがすでに (ボタン「<1>」と「<2>」に) 設定済みです。

- 数値ボタンを左クリックすると、そこに設定されているスクリプトを実行できます。また［ツール］メニューからスクリプトを選んだり、割り当てられたショートカットキー Ctrl+Alt+F#（# は 1 から 12）を使うことでも、同じようにスクリプトを実行できます。

デフォルトでは、スクリプトファイルは「scripts」フォルダーに格納されています。このフォルダーは、OmegaT のインストール先フォルダー（OmegaT.jar ファイルを含む）の配下にあります。

このフォルダーに新しいスクリプトファイルを追加すると、スクリプト ウィンドウの一覧に表示されます。

3. スクリプト言語

以下のスクリプト言語は、最初から使える状態になっています：

- Groovy（http://groovy.codehaus.org）：Java 仮想マシン向けの動的言語です。構文的に Java との互換性を最大限に保った上で、さらに Python や Ruby、Smalltalk に触発された強力な拡張機能を持ちます。

- JavaScript（JS と略すこともあり、Java とは無関係）：プロトタイプ ベースのスクリプト言語です。動的で、弱い型付け、ファーストクラス関数などの特徴を持ちます。オブジェクト指向、命令型、および関数型プログラミングのスタイルをサポートする、マルチパラダイム言語です。Firefox のような人気のあるプログラムで利用されていることから、とりわけオープンソースの分野で普及し好まれています。

すべての言語から、プロジェクトを先頭のオブジェクトとする OmegaT のオブジェクトモデルにアクセスできます。次に示すコードスニペットは、現在のプロジェクトから全ファイルの全分節を走査する groovy スクリプトの例です。このスクリプトは、もし訳文が存在していれば、その分節の原文と訳文を出力します：

```
files = project.projectFiles;
for (i in 0 ..< files.size())
{
    for (j in 0 ..< files[i].entries.size())
    {
        currsegment = files[i].entries[j];
        if (project.getTranslationInfo(currSegment))
        {
            source = currsegment.getSrcText();
            target = project.getTranslationInfo(currsegment).translation;
            console.println(source + " >>>> " + target);
        }
    }
}
```

付録G ウェブ上の OmegaT 情報

1. OmegaT のウェブサイトと SourceForge プロジェクトについて

OmegaT のウェブサイト [http://www.omegat.org/]には OmegaT に関するさまざまな情報へのリンクがあります。またボランティアによるユーザーサポートが、OmegaT Yahoo! ユーザーグループ [http://tech.groups.yahoo.com/group/omegat/]で提供されています。よくある質問（FAQ）ページ [http://tech.groups.yahoo.com/group/OmegaT/database?method=reportRows&tbl=1]はあなたの疑問に答えてくれる出発点になるでしょう。OmegaT の最新版については、www.omegat.org のダウンロード ページを参照してください。バグ報告 [https://sourceforge.net/p/omegat/bugs/]や機能改善リクエスト [https://sourceforge.net/p/omegat/feature-requests/]も行えます。

2. バグ報告

良いバグ報告には、次の 3 点が含まれていることを覚えておいてください：

- 再現するまでの手順
- その操作によって期待される動作
- 実際の動作

詳しくは、Joel Spolsky の『やさしいバグトラッキング（Painless Bug Tracking）』 [http://www.joelonsoftware.com/articles/fog0000000029.html]（日本語訳は http://japanese.joelonsoftware.com/Articles/PainlessBugTracking.html）を見てください。

開発者がバグを発見し、修正するのに助けとなる情報を添付してください。たとえば、該当ファイルやログの一部、スクリーンショットなどです。バグ報告や機能改善の要望は公開されますので、機密を含むファイルは追加しないでください。報告するバグの経過を知りたいときは、SourceForge ユーザー登録後、ログインしてから報告するか、報告ページ上部の Monitor をクリックしてください。

3. OmegaT プロジェクトに貢献する

OmegaT に貢献するには：

まずユーザーグループ [http://tech.groups.yahoo.com/group/OmegaT/]に参加してください（ウェブから登録するか、または OmegaT-subscribe@yahoogroups.com [mailto:OmegaT-subscribe@yahoogroups.com] 宛にメールを送信してください）。OmegaT の開発に協力したい場合は、開発者グループへ参加してください。ウェブから登録するか、または「subscribe」という件名のメールを omegat-development-request@lists.sourceforge.net [mailto:omegat-development-request@list.sourceforge.net?subject=subscribe] に送信してください。

OmegaT のユーザーインターフェースや取扱説明書、その他関連文書を翻訳するには：

まず OmegaT をあなたの言語に翻訳するには [http://www.omegat.org/en/howtos/localizing_omegat.php]をお読みいただき、OmegaT 翻訳者グループに参加してください（ウェブから登録するか、または「subscribe」という件名のメールを omegat-l10n-request@lists.sourceforge.net [mailto:omegat-development-request@lists.sourceforge.net?subject=subscribe] 宛てに送信してください）。

OmegaT プロジェクトを財政面から支援するには：

OmegaT の継続的な開発をサポートしたい場合は、こちらのリンクをクリックして OmegaT PayPal アカウント [https://www.paypal.com/cgi-bin/webscr?cmd=_s-xclick&hosted_button_id=9UB6Y2BBF99LL]へ寄付していただければ幸いです。

付録H ショートカットのカスタマイズ

1. ショートカットのカスタマイズ

メインメニューに表示されるほとんどの項目には、新しいショートカットを割り当てることができます。OmegaT の個人設定フォルダー（詳しくは個人設定ファイルを参照のこと）にショートカット定義ファイルを配置することによって、既存のショートカットを変更したり、新しいショートカットを追加することができるようになります。

ショートカット定義ファイルには必ず MainMenuShortcuts.properties という名前を付け、ショートカット定義を少なくとも一行は記述しておく必要があります。空行も含めることができます。また「//」で始まる行はコメント行として扱われます。「//」から行末までは、すべて無視されます。

MainMenuShortcuts.properties ファイルを修正した場合は、新しいショートカット設定を反映させるために、忘れず OmegaT を再起動してください。

ショートカット定義構文は、次の通りです：<メニュー項目コード>=<ショートカット>。<メニュー項目コード> の値は、後述する表を参照してください。<ショートカット> には割り当てたいキーの組み合わせを指定します[1]

<ショートカット> は次の形式に従います： 0 個以上の <修飾子>、続いて 0 個または 1 個の <イベント>、続いて 1 個の <キー>。ただし：

- <修飾子> は次の値を取ります：shift、control、ctrl、meta[2]、alt、altGraph
- <イベント> は次の値を取ります：typed、pressed、released
- そして <キー> には、キーボード上で利用可能な任意のキーを指定できます[3]。

例として、OmegaT のデフォルトのショートカット[4]は次のようになっています：

- projectOpenMenuItem=ctrl O
- editCreateGlossaryEntryMenuItem=ctrl shift G

最初のショートカットは（プロジェクトを）［開く...］用で、その次は［用語の追加］用です。

プロジェクトを開くために Shift+Ctrl+O を使いたいと思ったら、MainMenuShortcuts.properties ファイルを次のように修正してください：

projectOpenMenuItem=shift ctrl O

Mac 向けに［ツール］→［翻訳状況］を Shift+Command+S のショートカットで開けるようにしたければ、MainMenuShortcuts.properties ファイルに次の行を追加してください：

toolsShowStatisticsStandardMenuItem=shift meta S

ファイルを保存したら、OmegaT を再起動してください。新しいショートカットが、修正したメニュー項目の隣に表示されるはずです。システム用のショートカットと衝突しない限り、新しいショートカットは OmegaT で機能するでしょう。

---

[1]キーストローク（ショートカット）の完全な構文定義は、Oracle による次の Java 1.6 技術文書（ページの最後）を参照してください：Java 1.6 keystrokes shortcuts [http://docs.oracle.com/javase/jp/6/api/javax/swing/KeyStroke.html]
[2]Mac では、修飾子 meta の代わりに command キーを指定します。
[3]利用できるキーイベント（キー）の一覧は、Oracle による次の Java 1.6 技術文書を参照してください：Java 1.6 keyEvents description [http://docs.oracle.com/javase/jp/6/api/java/awt/event/KeyEvent.html]
[4]デフォルトの OmegaT のショートカット定義ファイルは Sourceforge から入手できます：デフォルトの OmegaT ショートカット [http://omegat.svn.sourceforge.net/viewvc/omegat/branches/release-2-6/src/org/omegat/gui/main/MainMenuShortcuts.properties]

Mac 向けのデフォルトの OmegaT のショートカット定義ファイルも Sourceforge から入手できます。それらはすべて「ctrl」の代わりに「meta」を使用しています：デフォルトの Mac 向け OmegaT ショートカット [http://omegat.svn.sourceforge.net/viewvc/omegat/branches/release-2-6/src/org/omegat/gui/main/MainMenuShortcuts.mac.properties]

2. プロジェクトメニュー

表H.1 プロジェクトメニュー

メニュー項目	デフォルトのショートカット	メニュー項目コード
新規作成...	Shift+Ctrl+N	projectNewMenuItem
開く...	Ctrl+O	projectOpenMenuItem
チームプロジェクトをダウンロード		projectTeamNewMenuItem
原文ファイルを追加...		projectImportMenuItem
MediaWiki から原文ファイルを追加...		projectWikiImportMenuItem
再読み込み	F5	projectReloadMenuItem
閉じる	Ctrl+Shift+W	projectCloseMenuItem
保存	Ctrl+S	projectSaveMenuItem
訳文ファイルを生成	Ctrl+D	projectCompileMenuItem
現在の訳文ファイルのみ生成	Shift+Ctrl+D	projectSingleCompileMenuItem
プロジェクト設定...	Ctrl+E	projectEditMenuItem
翻訳対象ファイル一覧...	Ctrl+L	viewFileListMenuItem
終了	Ctrl+Q	projectExitMenuItem

3. 編集メニュー

表H.2 編集メニュー

メニュー項目	デフォルトのショートカット	メニュー項目コード
取り消す	Ctrl+Z	editUndoMenuItem
やり直す	Ctrl+Z	editRedoMenuItem
参考訳文に置換	Ctrl+R	editOverwriteTranslationMenuItem
参考訳文を挿入	Ctrl+I	editInsertTranslationMenuItem
機械翻訳に置換	Ctrl+M	editOverwriteMachineTranslationMenuItem
原文に置換	Shift+Ctrl+R	editOverwriteSourceMenuItem
原文を挿入	Shift+Ctrl+I	editInsertSourceMenuItem
原文のタグを挿入	Shift+Ctrl+T	editTagPainterMenuItem
次の未使用のタグを挿入	Ctrl+T	editTagNextMissedMenuItem
選択領域を出力	Shift+Ctrl+C	editExportSelectionMenuItem
用語の追加	Shift+Ctrl+G	editCreateGlossaryEntryMenuItem
プロジェクトを検索...	Ctrl+F	editFindInProjectMenuItem
検索と置換...	Ctrl+K	editReplaceInProjectMenuItem
前の参考訳文を選択	Ctrl+↑	editSelectFuzzyPrevMenuItem
次の参考訳文を選択	Ctrl+↓	editSelectFuzzyNextMenuItem
参考訳文 1 を選択	Ctrl+1	editSelectFuzzy1MenuItem
参考訳文 2 を選択	Ctrl+2	editSelectFuzzy2MenuItem
参考訳文 3 を選択	Ctrl+3	editSelectFuzzy3MenuItem
参考訳文 4 を選択	Ctrl+4	editSelectFuzzy4MenuItem
参考訳文 5 を選択	Ctrl+5	editSelectFuzzy5MenuItem

メニュー項目	デフォルトのショートカット	メニュー項目コード
大/小文字設定→すべて小文字に		lowerCaseMenuItem
大/小文字設定→すべて大文字に		upperCaseMenuItem
大/小文字設定→Title case（単語の先頭を大文字）に		titleCaseMenuItem
上記を順に切り替え...	Shift+F3	cycleSwitchCaseMenuItem
既定値訳文として使用		editMultipleDefault
新しい訳文を登録		editMultipleAlternate
原文と同一訳文を登録		editRegisterIdenticalMenuItem

4. 移動メニュー

表H.3 移動メニュー

メニュー項目	デフォルトのショートカット	メニュー項目コード
次の未翻訳分節	Ctrl+U	gotoNextUntranslatedMenuItem
次の翻訳済み分節	Shift+Ctrl+U	gotoNextTranslatedMenuItem
次の分節	Ctrl+N または Enter または Tab	gotoNextSegmentMenuItem
前の分節	Ctrl+P または Ctrl+Enter または Ctrl+Tab	gotoPreviousSegmentMenuItem
分節番号を指定...	Ctrl+J	gotoSegmentMenuItem
次のメモ		gotoNextNoteMenuItem
前のメモ		gotoPreviousNoteMenuItem
分節の選択履歴を進む	Ctrl+Shift+N	gotoHistoryForwardMenuItem
分節の選択履歴を戻る	Ctrl+Shift+P	gotoHistoryBackMenuItem

5. 表示メニュー

表H.4 表示メニュー

メニュー項目	デフォルトのショートカット	メニュー項目コード
翻訳済み分節を色づけ		viewMarkTranslatedSegmentsCheckBoxM
未翻訳分節を色づけ		viewMarkUntranslatedSegmentsCheckBox
原文を表示		viewDisplaySegmentSourceCheckBoxMen
繰り返しのある分節を色づけ		viewMarkNonUniqueSegmentsCheckBoxM
メモ付き分節を色付け		viewMarkNotedSegmentsCheckBoxMenu
ノーブレークスペースを色付け		viewMarkNBSPCheckBoxMenuItem
空白文字を色付け		viewMarkWhitespaceCheckBoxMenuItem
双方向アルゴリズム制御文字を色付け		viewMarkBidiCheckBoxMenuItem
更新情報を表示 / 表示しない		viewDisplayModificationInfoNoneRadioBu
更新情報を表示 / 現在の分節で表示		viewDisplayModificationInfoSelectedRadio
更新情報を表示 / すべての分節で表示		viewDisplayModificationInfoAllRadioButton

6. ツールメニュー

表H.5 ツールメニュー

メニュー項目	デフォルトのショートカット	メニュー項目コード
タグ検証	Shift+Ctrl+V	toolsValidateTagsMenuItem
タグ検証（現在の文書のみ）		toolsSingleValidateTagsMenuItem
翻訳状況		toolsShowStatisticsStandardMenuItem
参考訳文の一致率		toolsShowStatisticsMatchesMenuItem
参考訳文の一致率（ファイル別）		toolsShowStatisticsMatchesPerFileMenuIt

7. 設定メニュー

表H.6 設定メニュー

メニュー項目	デフォルトのショートカット	メニュー項目コード
TAB キーで次の分節へ		optionsTabAdvanceCheckBoxMenuItem
終了時に常に確認		optionsAlwaysConfirmQuitCheckBoxMenu
機械翻訳		
用語ヒント / 用語ヒントを有効化		optionsTransTipsEnableMenuItem
用語ヒント / 完全一致		optionsTransTipsExactMatchMenuItem
入力補完→用語集...		optionsAutoCompleteGlossaryMenuItem
入力補完→定型文...		optionsAutoCompleteAutoTextMenuItem
入力補完→文字一覧...		optionsAutoCompleteCharTableMenuItem
字体...		optionsFontSelectionMenuItem
ファイルフィルター...		optionsSetupFileFiltersMenuItem
分節化規則...		optionsSentsegMenuItem
綴り確認...		optionsSpellCheckMenuItem
翻訳入力行...		optionsWorkflowMenuItem
タグ検証...		optionsTagValidationMenuItem
チーム...		optionsTeamMenuItem
外部 TMX ファイル...		optionsExtTMXMenuItem
表示...		optionsViewOptionsMenuItem
保存と訳文ファイル生成		optionsSaveOptionsMenuItem
プロキシ認証...		optionsViewOptionsMenuLoginItem
ウインドウを初期位置に復元		optionsRestoreGUIMenuItem

8. ヘルプメニュー

表H.7 ヘルプメニュー

メニュー項目	デフォルトのショートカット	メニュー項目コード
取扱説明書...	F1	helpContentsMenuItem
OmegaT について...		helpAboutMenuItem
最近の更新履歴...		helpLastChangesMenuItem

メニュー項目	デフォルトのショートカット	メニュー項目コード
ログ...		helpLogMenuItem

付録I 法律上の表示

1. 付属する文書について

著作権

OmegaT とともに配布される文書には、取扱説明書と readme.txt ファイルが含まれます。「5 分でわかる！ OmegaT の使い方」以外の文書は ©2013 Vito Smolej, ©2014 Vincent Bidaux の著作物です。「5 分でわかる！ OmegaT の使い方」は ©2005-2012 Samuel Murray の著作物です。

再頒布と改変

これらの文書は自由な文書です。あなたはこれを、フリーソフトウェア財団によって発行された GNU 一般公衆利用許諾契約書（バージョン 3 か、希望によってはそれ以降のバージョンのうちどれか）の定める条件の下で再頒布または改変することができます。

保証

これらの文書は有効活用されることを願って頒布されるものですが、その内容についてはまったくの無保証となります。商品性の黙示保証、または特定目的への適合性はまったく存在しません。詳しくは GNU 一般公衆利用許諾契約書を参照してください。

2. プログラムについて

著作権

OmegaT は © 2000-2014 Keith Godfrey, Zoltan Bartko, Volker Berlin, Didier Briel, Kim Bruning, Alex Buloichik, Thomas Cordonnier, Sandra Jean Chua, Enrique Estévez Fernández, Martin Fleurke, Wildrich Fourie, Phillip Hall, Jean-Christophe Helary, Thomas Huriaux, Hans-Peter Jacobs, Kyle Katarn, Piotr Kulik, Ibai Lakunza Velasco, Guido Leenders, Aaron Madlon-Kay, Fabián Mandelbaum, Manfred Martin, Adiel Mittmann, John Moran, Maxym Mykhalchuk, Arno Peters, Henry Pijffers, Briac Pilpré, Tiago Saboga, Andrzej Sawuła, Benjamin Siband, Yu Tang, Rashid Umarov, Antonio Vilei, Ilia Vinogradov, Martin Wunderlich and Michael Zakharov らの著作物です。

再頒布と改変

OmegaT は自由なソフトウェアです。あなたはこれを、フリーソフトウェア財団によって発行された GNU 一般公衆利用許諾契約書（バージョン 3 か、希望によってはそれ以降のバージョンのうちどれか）の定める条件の下で再頒布または改変することができます。

保証

OmegaT は有効活用されることを願って頒布されるものですが、その内容についてはまったくの無保証となります。商品性の黙示保証、または特定目的への適合性はまったく存在しません。詳しくは GNU 一般公衆利用許諾契約書を参照してください。

付録J 謝辞

1. みなさん、ありがとう！

現行のバージョンに存在する矛盾や記載漏れ、誤記などはすべて、筆者がその責を負うものです。この取扱説明書は、多くの人たちからの助けと支え無しには、存在し得ませんでした。とりわけ、次の方々に感謝します：

- Marc Prior へ。わたしの最初の下書きを校正してくれたとき、OmegaT と英語に対する彼の深い愛情を思い知りました。
- Didier Briel へ。彼の忍耐と、DocBook の厄介な問題に対する不断の助力が無ければ、本文書は完成しなかったでしょう。もちろん、リポジトリを完全な状態に維持するための、彼の注意深さと働きぶりは、言うまでもありません。
- Samuel Murray へ。初心者向けの章「5 分でわかる！ OmegaT の使い方」に対して。
- Will Helton へ。彼が下書きの最終チェックをしてくれたおかげで、多くの恥をさらす事態を免れました。彼の貴重な助けがなかったら、いったい幾つの前置詞 the や a が欠落していたことやら。
- Jean-Christophe Helary へ。わたしの知らない OmegaT の実行とコマンドライン引数、その他もろもろに関する簡潔な説明に、深く感謝します。
- 最後に、旧版の不備を見つけて OmegaT 文書管理システム [https://sourceforge.net/p/omegat/documentation/]に報告してくれた、すべての貢献者の皆さんへ。引き続き、よろしくお願いします！

索引

シンボル